Model Fitting for Motion Segmentation

Fernando Flores-Mangas

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Model Fitting for Motion Segmentation
Fernando Flores-Mangas
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto, 2015

The problem of rigid motion segmentation of trajectory data under orthography has been long solved for non-degenerate motions in the absence of noise. But because real trajectory data often incorporates noise, tracking failures, motion degeneracies and motion dependencies, recently proposed motion segmentation methods resort to non-trivial representations to achieve state of the art accuracies, at the expense of a large computational cost. This thesis proposes a method that dramatically reduces this cost (by two orders of magnitude) with minimal segmentation accuracy loss (from 98.8% achieved by the state of the art, to 97% achieved by our method on the standard Hopkins 155 dataset). Computational efficiency comes from the use of a simple but powerful motion model that explicitly incorporates mechanisms to deal with noise, outliers and motion degeneracies. Subsets of these motion models with the best balance between prediction accuracy and model complexity are efficiently ranked from a pool of candidates. Top scoring model combinations are then merged using an averaging technique to produce the final segmentation result.

To my wife.

Flaquita, you are the hero of this story.

Acknowledgments

Thanks to Allan$_+$ Jepson for his extensive support and guidance throughout my graduate studies. Allan's wisdom, humor, patience and strength will never be matched, and the balance between them is yet another critical piece I am still learning from him. A few lines will never express how grateful I am to you Allan.

Thanks to the members of my committee. Sven Dickinson, whose knowledge of the field, both deep and broad, together with his contagious enthusiasm always fueled brilliant conversations and sparked worthy ideas. David Fleet, whose attention to detail and stimulating comments were always really valuable. The combined feedback from both of you largely improved the quality and broadened the scope of my work.

Also thanks to Jim Little and Richard Zemel for agreeing to be my external examiners, and for the great feedback and questions that transformed my graduation process into a rewarding learning experience.

Thanks to my office mates over the years: Libby Barak, Marcus Brubaker, Paul Cook, Emily Denton, Nikola Karamanov, Varada Kolhatkar, Dustin Lang, Megana Marathe, Nona Naderi, Michael Reimer, Babak Taati and Matthijs van Eede (special thanks to you for being such an awesome friend/brother). Thank you all for the constructive environment, the laughs and the conversations.

Thanks also to all those other affiliated to U of T, including Laurent Charlin, Ady Ecker, Steve Engels, Francisco Estrada, Sam Hainoff, Midori Hyndman, Navdeep Jaitly, Michael Jamieson, Alex Levinshtein, Micha Livne, Jennifer Listgarten (special thanks for being a great friend, city guide and huge help during my first few years here), Diego Macrini, Chris Maddison,

Ted Meeds (special thanks for encouraging me to join TMU), Rolan Memisevic, Volodymyr Mnih, Ian Murray, Rama Natarajan, Mohammad Norouzi, Simon Osindero, Simon Prince, David Ross, Pablo Sala, Jake Snell, Ilya Sutskever, Kevin Swersky, Charlie Tang, Danny Tarlow, Graham Taylor and Jon Taylor, thank you all for the stimulating and lively atmosphere that you help built. Apologies to those I missed.

Thanks to all my friends, for giving me the ultimate gift of great memories.

Special thanks to my parents Fer y Lolita for all your support and encouragement throughout these years, and to my sister Any, for her kindness and love.

Finally, the most special thanks to Monica, for being my incredibly loving and hugely supporting wife, best fiend, coach, and companion. Flaquita, I will never be able to make up for the sacrifices you made. To you I dedicate this work.

Contents

Notation Index

F Number of frames in the sequence, 24
I Number of trajectories in $\hat{\mathbf{W}}$, 34
K Number of RANSAC trials, 35
M Number of candidate motion models in $\mathbb{C}$, 64
N Number of independent motions in the scene, 24
P Number of trajectories in the sequence, 24
T Number of closer-to-optimal labelings to be averaged, 85
$\mathbb{A}$ Tuple of F 2D Affine Transformations, 37
$\mathbb{C}$ Set of candidate local models of motion, 25
$\mathbb{E}$ Tuple of F Epipolar Direction Vectors, 38
$\mathbb{S}$ Tuple of all candidate model combinations, 25
$\mathbb{T}$ Tuple of N motion models, a.k.a. Model combination, 25
β_k Number of times trajectory k has been a model inlier, 66
σ Tuple of estimates of the magnitude of the noise, 38
λ_r Expected ratio of same-class trajectories within $\hat{\mathbf{W}}$, 35
$\hat{\mathbf{W}}$ Subset of spatially local trajectories, 27
$\mathbf{A}^f$ 2D Affine projection matrix for frame f, 37
$\mathbf{B}$ Binary matrix of inlier trajectories per frame, 38
$\mathbf{L}$ Binary matrix of motion-segmentation labels using one-hot encoding, 24
$\mathbf{M}$ Spatially local model of motion, 25
$\mathbf{P}$ Orthonormal basis for a motion model, 71
$\mathbf{W}$ Matrix of trajectory data, 24
$\mathbf{Z}$ Affinity matrix of trajectory labels, 32
$\mathbf{e}^f$ Epipolar direction at frame f, 38

1 Introduction

Computer based Motion Segmentation (MS) refers to the problem of grouping pixels, regions or trajectories into coherently moving sets. The solution to the MS problem is important not only because it enables higher level Computer Vision tasks, like 3D scene reconstruction or relative motion estimation (including the relative motion of the camera with respect to the world), but also because segmenting an image using independent motion information appears to be a strong cue for visual based scene understanding. In fact, it is well accepted that the human visual system relies on independent motion perception to perform tasks that include coherence detection, structure from motion [14], and even the interpretation of intentions and feelings [2], with very specific regions of the brain (see [2] for details) devoted to these rather complex perceptual tasks, many of which have been found to be indispensable to the interpretation of independent motion.

It has also been hypothesized that minimal stimuli, such as a dynamic dot display, is sufficient to discriminate between independently moving objects. A dynamic dot display presents the projections of a set of 3D locations from the different objects in the scene. The hypothesis was eventually confirmed [33], suggesting that sparse trajectory data often suffices to perceive motion independence. The principle was mathematically detailed for noiseless data several years ago [8], and the use of trajectory data is now the *de facto* standard within the MS community, motivated by the associated computational efficiency and the availability of methods that estimate trajectories from a moving scene by local image feature tracking [21, 31, 40, 44].

The process of going from a video sequence to a set of motion segmented

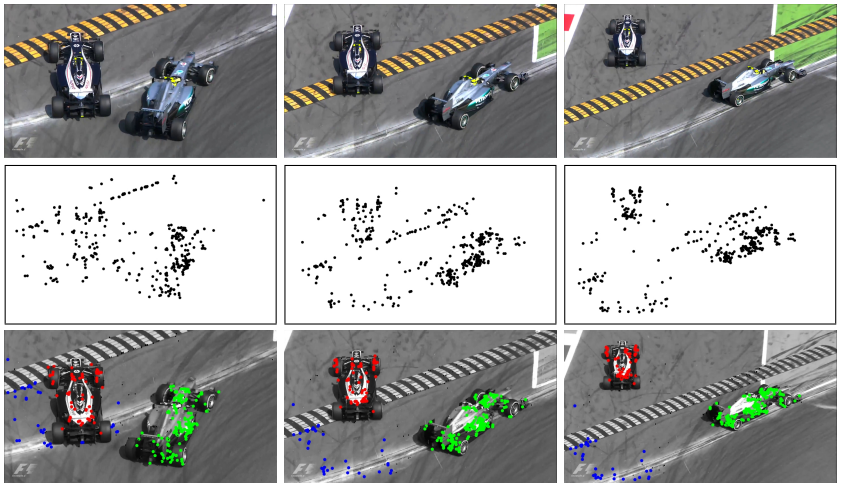


Figure 1.1: From image frames to motion-segmented trajectories. Three frames from a sequence from the `Formula 1` dataset. Original images at the top row, trajectories in the middle and segmented results at the bottom row.

trajectories is described with Figure 1.1. The top row of shows three of the original frames. A feature tracking algorithm then detects [40] and tracks [31, 44] salient points across all frames. The middle row shows the resulting 354 trajectories that survived the entire sequence which, together with the number of independent motions, becomes the input to the MS algorithm. The bottom row shows color-coded trajectories, according to the output labeling produced by a MS algorithm.

For much of this thesis, a segment is a set of trajectories that is consistent with the affine projection of 3D rigid motion. In 1.1, the segments correspond to the three sets of trajectories consistent with different rigid motions.

The problem of MS of trajectory data is fundamental to the Computer Vision community. A large number of contributions already exist, some that have increased the understanding of the theory behind this problem, and some others that have introduced methods that exploit the existing understanding to provide solutions to the MS problem. The contributions of this thesis include an improved solution that is highly accurate and that requires a fraction of the computation time used by existing methods. Experimental results also showed that the proposed method is also applicable to a wide range of situations, including objects that slightly deform and sequences with many outlier trajectories. Computational efficiency is achieved by combining the use

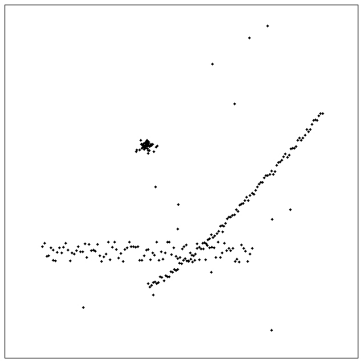


Figure 1.2: Toy problem's input 2D data. Two linear models and one punctual model for $N = 3$.

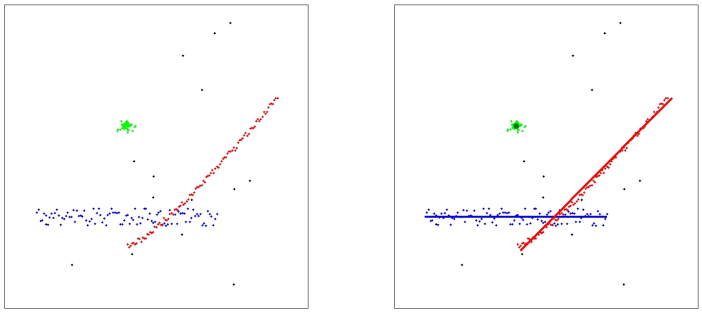
of an efficient model fitting technique, together with the common assumption of local coherence. Applicability to a wider range of problems is possible thanks to the built-in robustness and the ability to deal with varying levels of trajectory noise.

Before proceeding with the rest of the thesis, we think it is beneficial to introduce a Toy Problem (TP) that is similar to MS in many of the most critical aspects, but unlike the MS problem, can be drawn in 2D and may in fact already evoke some intuition from the reader. This TP will also help introduce the key difficulties of doing MS on non-trivial datasets.

1.1 Toy Problem

Assume $\mathbf{W}$ is a $2 \times P$ matrix that contains the 2D locations of p points that originate as the noisy observations of a series of independent linear or punctual phenomena, plus some outlier points. The toy problem consists on finding subsets of points that are coherent with a set of linear or punctual models, assuming that the number of independent models N is known a priori.

Figure 1.2 shows a plot of an example $\mathbf{W}$ with $p = 270$ points and $N = 3$ independent sources: two linear and one punctual. Clustering these points into three independent classes is relatively clear from visual inspection (Figure



(a) (b)

Figure 1.3: Toy problem. a) Ground truth labeled data. b) Underlying models: linear (blue and red line segments) and punctual (dark green dot).

1.3.a shows the ground truth labelling), except perhaps for points closest to the intersection of the two linear phenomena, as well as for outlier points (shown in black in Figure 1.3.a). However, visual inspection also reveals two important aspects of the data. First that the magnitude of the noise is different depending on the underlying class for each 2D point, and second, that in some cases the data does not really align with a linear or a punctual model. In particular, look at Figure 1.3.b, where the least-squares linear model fit to the red-class points reveals that the true underlying phenomenon is very likely non-linear. These and other issues are shared by the MS problem. A more detailed list is presented next.

1.1.1 MS $\simeq$ TP

The MS problem is similar to the TP in the following critical aspects.

1. Models of different capacities (*i.e.*, different number of free parameters) must be considered when fitting different subsets of trajectories due to the common presence of motion degeneracies (cameras that only rotate, planar objects, etc.). In the TP, linear and punctual models are available, and certainly the linear model allows for one extra degree of freedom, compared with the punctual model.

2. Motion dependencies (like the joints of an articulated object) create situations where it is more difficult get an estimate of the magnitude of

the noise, and where the uncertainty about the resulting segmentation labeling grows. In the TP, the issue appears at the intersection of two models, like the two linear models of the example.

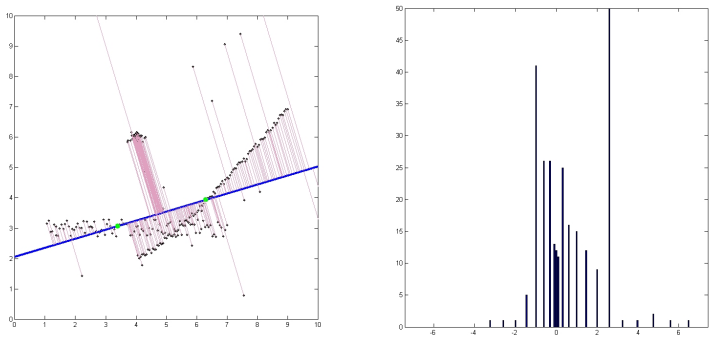
3. The average magnitude of the noise is unknown and can be different for different groups of trajectories due to varying imaging conditions (loss of image contrast, illumination changes, etc.), as well as motion properties (motion blur, deforming textures, etc.). The magnitude of the noise for each point in the TP is a function of the underlying class.

4. Unaccounted image formation processes (perspective effects, barrel distortion, etc.) can result in trajectories whose motion deviates from the model's predictions. In the TP, the distribution of trajectories may only be approximately linear (or punctual).

5. Some trajectories cannot be explained by any of the models (drifting trajectories, tracking errors, etc.). Some 2D points of the TP are also away from any of the true underlying models.

1.1.2 Solving the TP

A standard solution to the problem of robustly fitting a set of models to the toy problem's 2D data would be to use the Random Sampling and Consensus (RANSAC) algorithm, which iterates three steps until most of the data has been explained (or the maximum number of models have been reached): first, fit a large number of models using the minimal number of randomly sampled control points. Second, identify the model with the largest number of data inliers. Third, refine the model estimate and remove the resulting set of inliers from the dataset. At first glance, RANSAC appears as a sensible option since it is specifically designed to deal with outliers, it can be stopped after the first N subsets of inliers have been found, and it is computationally very efficient. However, the standard RANSAC does not naturally extend to problems where different types of models are necessary to efficiently describe the data, and even if we assumed it did, RANSAC is by design a greedy algorithm, meaning that the decision of keeping or discarding a model is purely based on the (local) evidence of each model. In other words, RANSAC fails to incorporate the knowledge of how a particular model interacts with the other $N-1$ models to jointly describe the data.



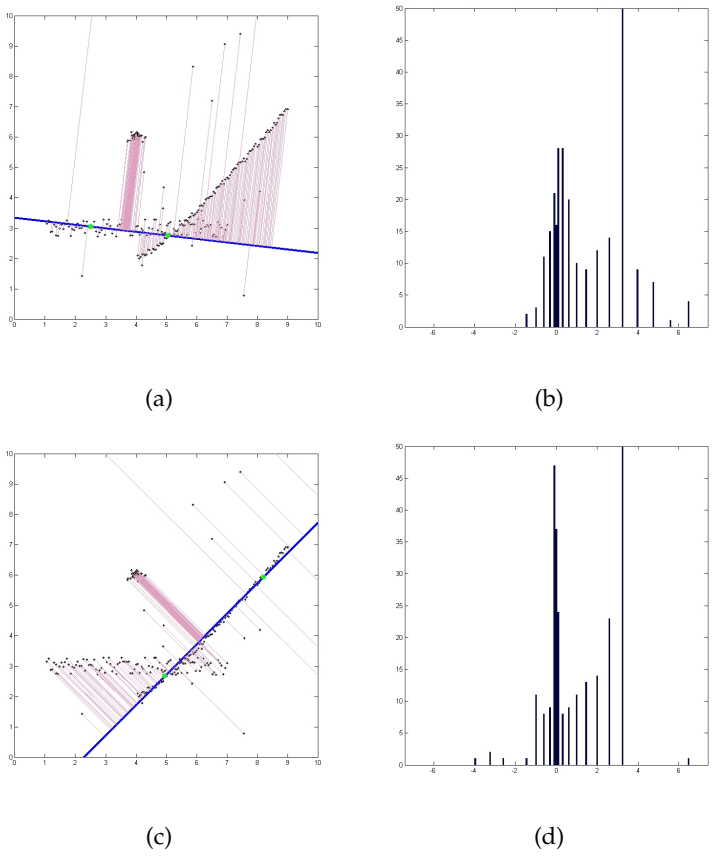
(a) (b)

Figure 1.4: A spurious candidate model. a) Observed data with overlaid model (blue line) and residuals (red lines). b) Histogram of signed residual magnitudes. Note presence of may residuals of small magnitude.

A typical conundrum that originates from the greedy nature of RANSAC occurs in situations like the one depicted in Figure 1.4.a, where the two control points (in green) of a linear model are drawn from two independent sources, rendering a model that coincidentally describes the data of both linear phenomena with not very large residual magnitudes (red lines). Figure 1.4.b shows a histogram of distances to the line. Distances are signed to reflect which side of the line each point lies on. It is at this stage when RANSAC must determine whether this histogram (or any other criterion based on the residuals) is one of a valid model or not. Typically, the final call is based on a dynamically determined inlier threshold (a fixed threshold is unfeasible for data where the magnitude of the noise changes between classes). We argue that this is a very difficult task, especially when only local evidence from a single model is available.

Dynamic Estimation of the Magnitude of the Noise

The difficulty of the problem is only fully exposed when comparing the histogram of the contaminated model of Figure 1.4.b with the equivalent histograms of two clean model candidates that actually represent the underlying phenomena (Figure 1.5.b and 1.5.d). Clearly, the later histograms show a higher density of small magnitude residuals, but the correct magnitude of the noise is certainly not obvious from this evidence alone.

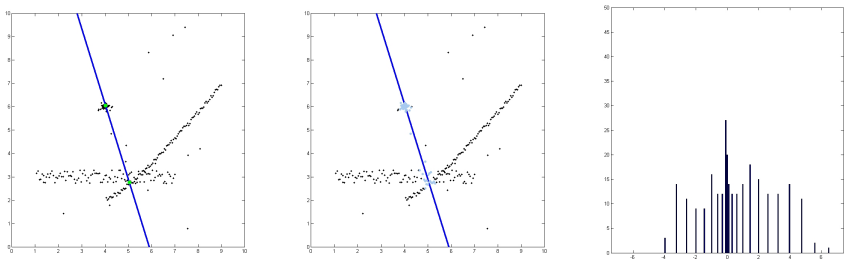


(a) (b)

(c) (d)

Figure 1.5: Two candidate models that align with the underlying linear phenomena. Left: observed data with overlaid model (blue line) and residuals (red lines). Right: histogram of signed residual magnitudes. Note that the histogram shown in b) has no obvious gap between the magnitudes of the inlier and the outlier residuals.

One of the key contributions of this thesis is an algorithm in which the estimation of the magnitude of the noise of each model can be postponed to a stage when the residuals of all N models are already available. This allows simultaneous characterization of the residuals of all models and all trajectories while also being able to determine the inlier subsets for each model. To achieve this goal we propose an efficient mechanism that finds the best performing model combinations from within a potentially very large set.



(a) (b) (c)

Figure 1.6: A spurious linear model that explains the data of a punctual phenomenon. a) Control points (green) and fitted model. b) Inlier subset using an arbitrary threshold. c) Histogram of signed residual magnitudes.

Model Selection

In the context of the TP punctual and lineal models are used to explain the data in **W**, but choosing the right model type is a non-obvious issue. A typical problem occurs when the control points of a more loosely constraining model (in this case a line model) coincidentally fits data from a more constrained phenomenon (a punctual one) as shown in Figure 1.6.a. The model will explain the punctual phenomenon's data well, together with some more data from independent sources, as shown in Figure 1.6.b, rendering a contaminated model. In addition, it is possible that sufficient support is found to keep the model, given the large density of residuals with small magnitude (Figure 1.6.c).

Another contribution of this thesis is a mechanism that detects if a more constraining model is sufficient to describe a subset of the data, without fitting the less constraining one. The mechanism looks at the spatial distribution of the residuals of the more constraining model and classifies it as sufficient or not. The classification is formally based on epipolar geometry.

The overall strategy is to propose many individual models of varying capacity, and then consider the most promising combinations of N models, according to how accurately and efficiently they explain all the data. We use (several of) these most promising N-model combinations to estimate the final segmentation result.

2 Related Work

This chapter introduces the most relevant contributions in the field of MS. The goal is to understand the relationship between different algorithms, as well as to identify strengths and weaknesses. A taxonomy that facilitates the classification of all these algorithms is proposed. Most of the related work described here deals with segmentation of trajectories from rigid objects under orthography. The ones that deviate from these assumptions are explicitly noted but still included for completeness and to understand the difficulties.

A common notation for the problem of Rigid Motion Segmentation is described next.

2.1 Notation

Let $\mathbf{p}_i = [x_i^w, y_i^w, z_i^w]^\top$ be the coordinates of a point in 3D space. If a camera moves around this object (or if the camera is fixed and the object moves), then the projection of $\mathbf{p}_i$ at frame f can be computed in the following two steps. First the point $\mathbf{p}_i$ is transformed into camera coordinates via the following Euclidean transformation:

$$\begin{bmatrix} \mathbf{s}_{fi} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_f & \mathbf{t}_f \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} = \mathbf{M}_f \mathbf{s}_i, \tag{2.1}$$

where $\mathbf{R}_f$ and $\mathbf{t}_f$ correspond to the rotation and translation components of the motion for the f^{th} frame, respectively. This transformation is commonly referred to as the External Calibration Matrix.

In step two, assuming a scaled orthographic camera model, image coordi-

nates can then be computed with the following linear transformation:

$$\begin{bmatrix} x_{fi} \\ y_{fi} \end{bmatrix} = s_0 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{s}_{fi}, \tag{2.2}$$

where s_0 is a constant scale factor. Using a scaled orthographic projection is useful, as a linear approximation to the true underlying non-linear perspective projection, although it only holds true for objects with small relative depths that are far from the camera. This transformation is typically referred to as the Internal Calibration Matrix.

Most papers use a compact representation for the projections of all points in all frames. This is achieved by stacking the x_{fi} and y_{fi} coordinates of all F frames into column vectors and then grouping all those P vectors into a $2F \times P$ matrix, as in:

$$\mathbf{W} = \begin{bmatrix} x_{11} & \cdots & x_{1P} \\ y_{11} & \cdots & y_{1P} \\ \vdots & \ddots & \vdots \\ x_{F1} & \cdots & x_{FP} \\ y_{F1} & \cdots & y_{FP} \end{bmatrix}. \tag{2.3}$$

This matrix is typically referred to as the observation matrix. The goal of the motion segmentation problem is to group trajectories (columns of $\mathbf{W}$) according to the independent motion of the objects in the scene.

2.2 Taxonomy

In an attempt to structure the presentation of the existing contributions, a taxonomy based on the main underlying principle of each method is proposed. Several classes of practical solutions were identified, such as Factorization-based methods, or Random Sampling based methods, to name two. Notice that some methods do not perfectly align with any of the proposed classes, and others do so to more than one. While most papers introduce a practical (or algorithmic) solution to the problem of MS, some others aim to increase the theoretical background or the understanding of the problem without an implementation of the proposed principles. A different class is included for these theoretically inclined papers.

2.2.1 Matrix factorization

Matrix factorization methods are based on the fact that an observation matrix can be decomposed into motion and structure matrices using standard matrix factorization techniques. Some of the theory behind these methods is presented here as it provides intuition regarding the nature of the problem. These were some of the first practical motion segmentation methods.

Tomasi and Kanade first presented the underlying theory behind the earliest motion segmentation algorithms in 1990. Their paper [45] shows that in the absence of noise and under orthography, the observation matrix is highly rank deficient. In fact, for a single rigidly moving object, one can decompose an observation matrix into:

$$\begin{bmatrix} x_{11} & \cdots & x_{1N} \\ y_{11} & \cdots & y_{1N} \\ \vdots & \ddots & \vdots \\ x_{F1} & \cdots & x_{FN} \\ y_{F1} & \cdots & y_{FN} \end{bmatrix}_{2F\times P} = \left[\begin{array}{c|c} \mathbf{i}_1^T & t_{x1} \\ \mathbf{j}_1^T & t_{y1} \\ \vdots & \vdots \\ \mathbf{i}_F^T & t_{xF} \\ \mathbf{j}_F^T & t_{yF} \end{array}\right]_{2F\times 4} \begin{bmatrix} s_1 \dots s_N \\ \mathbf{1} \end{bmatrix}_{4\times N} = \mathbf{MS} \quad (2.4)$$

where vectors $\mathbf{i}_f^\top$ and $\mathbf{j}_f^\top$ are the first two rows of the f^{th}-frame's rotation matrix, and t_{xf} and t_{yf} are the x and y components of the translation. By inspection, Equation 2.4 shows that $\mathbf{M}$ and $\mathbf{S}$ are of at most rank 4, which then implies that $\mathbf{W}$ is at most rank 4. This rank constraint limits the dimensionality of the subspace in which trajectories from a single rigid object lie, enabling motion segmentation via Matrix Factorization.

By means of singular value decomposition (SVD) one can factorize $\mathbf{W}$ as:

$$\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^\top = (\mathbf{U}\Sigma^{\frac{1}{2}})(\Sigma^{\frac{1}{2}}\mathbf{V}^\top) = (\mathbf{U}\Sigma^{\frac{1}{2}}A)(A^{-1}\Sigma^{\frac{1}{2}}\mathbf{V}^\top) = \mathbf{MS}, \quad (2.5)$$

which leads to: $\mathbf{M} = \mathbf{U}\Sigma^{\frac{1}{2}}\mathbf{A}$ and $\mathbf{S} = \mathbf{A}^{-1}\Sigma^{\frac{1}{2}}\mathbf{V}^\top$. Constraints can be derived to find the matrix $\mathbf{A}$, given that paired rows of $\mathbf{M}$ ($\mathbf{i}_f$ and $\mathbf{j}_f$) must be orthogonal and of equal length.

Years later, Costeira and Kanade [8, 9] reported that when $\mathbf{W}$ contains two independently moving objects, the recovered shape is, in general, a linear combination of the subspaces of the two motions, but their main contribution

was the introduction of the Shape Interaction Matrix $\mathbf{Q} = \mathbf{V}\mathbf{V}^\top$, with $\mathbf{V}$ equal to the singular vectors associated to non-zero singular values (from the SVD: $\mathbf{U}\Sigma\mathbf{V}^\top = \mathbf{W}$). This matrix provides information about whether trajectories belong to the same rigid object, regardless of the underlying structure [9].

The work from Boult and Brown [4] deepens the analysis of the rank of $\mathbf{W}$. In particular, they study the effects of linearly dependent motions and propose a segmentation algorithm based on the rank of subsets of the observation matrix created by clustering the columns of $\mathbf{V}$ (the same $\mathbf{V}$ as above). Their algorithm uses two constraints. First, subsets of independent objects must have rank lesser than four, and second, the combined rank of all the subsets should be equal to the rank of the entire observation matrix.

In the early 2000s, Zelnik-Manor and Irani [62] observe that the shape interaction matrix of two objects will have a block diagonal only when both motions are linearly independent, otherwise, the off-diagonal blocks are non-zero. The solution they propose involves first computing the inner product of all trajectories: $\mathbf{Q} = \mathbf{W}^\top\mathbf{W}$, since the angles between trajectory vectors of points from the same object should be, on average, smaller than those between trajectories of different objects. Second, they compute a similarity matrix of trajectories using a subset of the most dominant eigenvectors $\mathbf{V}$ of Q using $\hat{q}_{ij} = \sum_k e^{-\|v_k(i)-v_k(j)\|^2}$, hoping that the within-object variation would be removed by the rank deprivation. They demonstrate that $\hat{Q}$ is (almost) block diagonal even for partially dependent motions.

Bregler, Hertzmann and Biermann [5] extend the idea of matrix factorization of Costeira and Kanade and apply it to non-rigid objects to factorize the observation matrix $\mathbf{W}$ into a weighted motion basis matrix. The idea is that a generative model of a particular structure can be a linear combination of basis structures. The assumption is that only a finite number of linear deformations ever occur. Once the basis structures are computed, their weights can be varied to observe the modes of deformation. As with all other factorization methods, the effects of noise are destructive and even a few outliers can quickly overweight the cohesive information of many inliers. Llado, Del Bue and Agapito [30] extend this idea to non-rigid 3D factorization, but now from perspective images, with a very similar approach.

Observations All these methods are of great theoretical relevance and understanding them provides great intuition on the motion segmentation problem.

Their computational feasibility makes them particularly attractive, and so does their mathematical elegance. On the other hand, since most real world feature tracking algorithms produce noisy trajectories and are prone to generate outliers, most of these algorithms are, by themselves, of restricted practical use. In the presence of noise, shape interaction or affinity matrices will be mainly composed by non-zero entries, making segmentation non-trivial. Degenerate motions or degenerate surfaces might be merged and jointly described, and outliers often lead to contaminated motion models. Most of these drawbacks are due to the immediate use of all input trajectories to generate motions without any analysis. In other words, the underlying constraint used here is the linear independence between trajectories of differently moving objects, but because the linear subspaces are created without reasoning about the structure of the noise or the presence of outliers, or without considering that partial dependency may occur, these methods are of limited practical use.

2.2.2 Subspace separation

These algorithms are similar in essence to the matrix factorization ones, but in these contributions, the subspaces that explain the origin of individual trajectories are built prior to doing segmentation and, in some cases, acknowledge the presence of noise and existence of outliers.

The method proposed by Fan, Zhou and Wu [11] came first (2004). They build upon the theory of independent motions lying within orthogonal subspaces, presented by Wu, Zhang, Huang and Lin [57], where the span of a set of trajectories is characterized using a projection matrix. With this representation, both the distance between two subspaces and the membership of a trajectory to a particular subspace can be easily estimated. Their contributions include an objective function that encourages orthogonality between all shape subspaces, and at the same time, evaluates the goodness of fit for each trajectory to its preferred model. The optimization of the objective function is done using a genetic algorithm method and a special case deals with partially dependent motions (that are non-orthogonal).

Yan and Pollefeys [59] have a number of contributions on the understanding and development of the theory on articulated motions (see Section 2.2.5 for a random-sampling based contribution). In this paper, the authors build motion subspaces by first projecting all the trajectories into a low-dimensional

space and then fitting a manifold to the neighborhood of each point. The argument is that most neighboring points lie on a nearby underlying subspace. Clustering is then performed on an affinity matrix built using a distance metric between the estimated local subspaces of each pair of points.

The work from Rao, Tron, Vidal and Ma [35] aims at solving the problem of MS in the presence of outlier trajectories (from non-rigid objects, or from randomly moving or drifting tracks), as well as incomplete trajectories, where the position of the feature point is not available for all frames. The problem is formulated as a robust subspace separation where a matrix is partitioned into sub-matrices such that each sub-matrix is maximally rank deficient. This is achieved by minimizing a surrogate function to the rank of a matrix, using the general purpose segmentation by coding and compression algorithm from [32].

The work from Zappella, Llado, Provenzi and Salvi [60] is an upgrade to the Local Subspace Affinity (LSA) work of Yan and Pollefeys [59], pointing out that LSA could use two improvements: the first one is the ability to work with incomplete trajectories, and the second one is the possibility to automatically estimate the main parameter in LSA, which has to do with the magnitude of the noise and is used to determine the effective rank of the observation matrix. Automatic estimation of this parameter is achieved by maximizing the entropy of an affinity matrix that is computed from the rank-limited reconstructions of the trajectory data. Please refer to Section 7.3 for details on estimating the number of motions, as this contribution is more relevant to that topic than to the original problem of MS.

The method by Elhamifar and Vidal [10] uses the very simple but powerful principle of sparse representation. In their method, every trajectory is represented as a weighted linear combination of a very limited set of other trajectories. The weights then are used as the entries of an affinity matrix which is, in turn, spectral-clustered to obtain the final segmentation result. This method is one of the most cited recent MS algorithms, partly because of the elegance of the approach, but also because the reported segmentation accuracy was almost 3 times better than the state of the art at the time of publication. The method is robust to outliers and handles motion degeneracies well, however, estimating the sparse representation is a computationally expensive process.

Observations One of the most noteworthy contributions of these papers is simply acknowledging that the inner product of trajectories (the angle between the two) fails to reflect the true geometric constraints of multi-view geometry, but that accurate subspace modelling does, including that of partially dependent motions. Another message from the papers in this section is about the potentially large benefit of finding robust ways to estimate the underlying subspaces and their effective dimensionality, particularly when solving the MS problem in its most general context (*i.e.*, under the presence of motion degeneracies, structured and unstructured noise, outliers, etc.).

2.2.3 EM-based methods

Expectation Maximization (EM) is an iterative algorithm used to find (locally optimal) maximum likelihood estimates of parameters in probabilistic models. In the context of motion segmentation, EM typically alternates between estimation of statistics of the model parameters (E-step) and trajectory assignments (M-step).

Vasconcelos and Lippman [52] proposed one of the first EM algorithms for motion segmentation. Their algorithm works directly on image pixels. They model motion as realizations of a stochastic process characterized by a Gaussian mixture density with as may components as there are motions. Model assignments are the hidden variables. The goal is to find values for the motion parameters that maximize the likelihood of the observed data. Their likelihood model includes a goodness of fit term and there is also a spatial coherence prior implemented with a Markov random field. Since the algorithm works at the image level, no notion of structure can be incorporated, and therefore only 2-D affine motions are modeled.

Gruber and Weiss [16, 17, 18] exploit the benefits of a probabilistic model representation to enhance motion segmentation solutions by including temporal and spatial coherence prior models. Their approach resembles techniques based on factorization of an observation matrix ($\mathbf{W}$) but under a factor analysis framework. Temporal coherence is introduced by modelling the first (speed) and second (acceleration) derivatives of the location of a point at each frame. The spatial coherence prior is based on the assumption that neighboring points should belong to the same motion model and is enforced using a 2-D graphical model. Maximum likelihood motion parameters are estimated using

EM.

Sugaya and Kanatani [42] acknowledge and address the issue of degenerate motions by dealing with them first, using a special EM algorithm, and then extracting the general 3-D ones using a more general EM algorithm. They call this procedure multi-stage learning. The degenerate motion learning process assumes all motions are in a 2-D affine space allowing for planar motions and changes in scale only. The resulting segmentation is processed by the second stage, which in turn uses a 3D affine model. The claim is that, if motions are really degenerate, the solution found by the first stage (with degenerate constraints) will not be modified (or affected) by the second stage (with non-degenerate constraints).

A probabilistic model is also proposed by Lakdawalla and Hertzmann [27]. Their formulation is quite simple. Describe camera parameters, structure, motion, texture and lighting with one big likelihood model and optimize to obtain a maximum a posteriori estimate. They assume constant internal camera parameters, but they estimate external camera parameters for each frame. Texture is modelled with RGB using the Lambertian lighting equation, which also models the direction of light. Smoothness priors for surface, rotation, translations and scaling regularize the model. Conjugate gradient is used to optimize (for over a week).

Observations The graphical model representation of these algorithms allow the use of alternative and complementary sources of information. Prior models can be naturally included as well. On the downside, iterative (EM-like) algorithms converge to a local optimum that strongly depends on the quality of the initial guess.

2.2.4 Algebraic methods

The following algorithms are motivated by mathematically elegant properties that arise due to the nature of the motion segmentation problem.

The work of Wolf and Shashua [56] is a great example as they tackle the problem of two-body segmentation from two-frame trajectory data under perspective projection using only expressions elegantly derived from the Epipolar Constraint $p'\mathbf{F}p = 0$. Their paper introduces a way to non-linearly represent the simultaneous epipolar constraint of two objects (as the product

of two single object ones), and based on this equation they derive a series of properties that provide information about the segmentation label of each trajectory, the location of the Epipoles and the number of trajectories necessary to estimate all these unknowns.

Vidal and Hartley [53] propose an algebraic method for motion segmentation of point trajectories under all kinds of projections for both non-degenerate and degenerate motions. Their method requires projecting each trajectory onto a 5 dimensional space, arguing that in order to segment motions, it is enough for them to be different along one dimension alone. Then, a 5-dimensional polynomial of degree n is fitted to the data, with n equal to the expected number of motions. Their motivation is that a set of n hyper-planes can be represented by the product of n linear polynomials, one for each plane. Then the derivative of this polynomial is evaluated at each point, as it provides the normal of the hyper-plane in which it lies. A similarity matrix is populated using a distance metric between the normals of each pair of points. Spectral clustering is then used to do the segmentation.

The method proposed by Rao, Yang, Wagner and Ma [36] is the quadratic extension of Vidal and Hartley's work. They argue that the hyper-plane first derivatives are not enough for classification since most of the motion constraints (epipolar or homography) are of quadratic nature. Therefore gradients, Hessians and tangents are analyzed as segmentation features in a very similar framework.

Goh and Vidal's work [15] is based on an existing embedding technique called Locally Linear Embedding which essentially provides non-linear dimensionality reduction, from which features for each trajectory are then computed. Segmentation is achieved by clustering these features using k-means.

Observations Despite the mathematical elegance of these methods, some of them create very large systems of equations that appear highly unstable to noise. Even the authors themselves discuss the destructive effects of noise or outliers in the data. Still, these methods provide some intuition with respect to the relatively low-dimensional manifold in which the trajectories of each object lie.

2.2.5 Random sampling methods

Some authors acknowledge that existing tracking algorithms are not ideal. Due to appearance or illumination changes, motion blur or other imaging artifacts, automatically generated tracks have inaccuracies (noise) or completely drift from one frame to another (outliers). The solutions in this section include sampling mechanisms that provide robustness against these issues.

Torr's work [48] provides the aforementioned robustness, but he is also one of the first to acknowledge the issue that degenerate motions (i.e. pure camera rotations) or degenerate shapes (i.e. planes) require different motion models than their general 3-D counterparts, and he addresses it. His approach is based on the study of model capacity. The result is the so-called geometric robust information criterion (GRIC) that balances between a robust metric for goodness of fit versus the dimension and total number of parameters of the motion model. The proposed segmentation algorithm has two stages. First, candidate motions are instantiated using RANSAC for both homographies (2-D) and fundamental matrices (3-D) and their GRIC score is computed. The best model is kept and its corresponding matches removed, repeating until all matches are exhausted. The second stage involves assigning samples to candidate motion models. A cost function that accounts for the goodness of fit, the dimension and the number of parameters of the model is minimized to obtain the segmentation.

Schindler and Suter propose a couple of very similar, theoretically sound, random sampling based methods. Their first paper [37] deals with two-view structure and motion estimation, their following [38] extends the work to full trajectories, but they are both very similar. One of the key features is acknowledging that the magnitude of the noise should be estimated to accurately distinguish inlier subsets. They also mention that an outlier model is necessary. Similarly to Torr, the proposed algorithm has two stages. The first one generates candidate models using RANSAC, also instantiating homographies and fundamental matrices. The difference with respect to Torr's is that the magnitude of the noise is also estimated and likelihoods are assigned to determine an inlier set, keeping only motion models with a reasonably large number of inliers. The second part of the algorithm deals with the assignment of points to candidate motion models. To that end, they propose an objective function that balances the goodness of fit with the complexity of

the model, which is then optimized to find the most likely segmentation. A sound, information theory-based framework is used in [38] to motivate and solve the problem of model selection.

Tong, Tang and Medioni [46] deal with the problem of motion segmentation from potentially mismatched features between image pairs. The proposed method uses 4-D tensor voting to estimate a subspace manifold using a smoothness prior. Once the manifold is estimated, the membership probabilities of each trajectory to each manifold can be computed. These probabilities are then used to influence the sampling probabilities of a RANSAC method to generate the final set of motion models.

The early work of Yan and Pollefeys on motion segmentation [58] is focused on describing the representation of motions that originate from articulated objects. The idea is that the performance of RANSAC can be improved by using priors that influence the frequency each point is used to estimate models. The prior model is built by computing the inner products between pairs of trajectories. The intuition is that the least shared subspace, the lesser the value of the inner product. The remainder of the method is similar to all the other RANSAC-based approaches, and unfortunately, the articulated motion segmentation theory is not further exploited in the paper.

Li [29] proposes a mixture of fundamental matrices method that works almost the same way a mixture of gaussians does. First fundamental matrices are instantiated using RANSAC and the 8-point algorithm. Following, membership probabilities for each point to a motion model (or fundamental matrix) are computed by minimizing an objective function that includes a metric for goodness of fit and the complexity of the entire model.

The work from Laptev, Belongie, Perez and Wills [28] aims to exploit periodic motion for segmentation. Their approach is limited by a series of restrictive assumptions (constant camera translation, constant object translation, etc.), but it is included here as the approach also uses the idea of recovering motion models, as well as the frequency of the periodic motion, using vanilla RANSAC.

Observations Random sampling techniques are a reasonable way to accommodate outliers, which is one of the common features of all these papers. However it is now clear that that the field of motion segmentation is notoriously more mature by the time of publication of some of these papers, as the

authors clearly understand the true issues behind the problem: the need of multiple types of models, online estimation of the noise, and the possibility of partially dependent models, to name just a few. Without a doubt, a robust solution must have all these elements. Moreover, authors start tackling the problem of multiple plausible interpretations of the same data, using general model selection criteria.

2.2.6 Planar approximations

The following contributions deal with recovery of planar motion avoiding the problem of recovering structure . They are presented here to acknowledge their existence but are considered outliers from this review. First is the work by Kumar, Torr and Zisserman [26] who present a method where motion segmentation is achieved using a layered model. The algorithm first learns 2-D templates of the projections of rigid structures and then uses them to compute their most likely location at each frame. Since no 3-D model is built, only fronto-parallel rotations are modelled.

Briassouli and Ahuja [6] propose a Fourier transform (FT) based motion segmentation technique. They argue in favor of using FT due to its illumination invariance properties. Quotients of FT coefficients between frames are used to estimate translation. Then FT coefficients are mapped to a polar coordinate system to estimate rotations. Again, off-plane rotations will produce inaccurate results.

Observations The planar assumption is too restrictive to do general motion segmentation, although when used in small, local patches, it may provide a good initial guess to other more robust methods.

2.2.7 Theory papers

The papers presented below are important because of the relevance of their theoretical contributions towards the understanding of the field over their practical methods, or for historical reasons.

The work of Ullman [50] is an example of both. The paper explains some of the first technical details known about the recovery of 3-D structure from 2-D images. At the same time it shows how computer vision from the early 80s was much more related with human perception (compared to today), as the visual

system was used to motivate and explain methods and even to evaluate results from computer vision research. One of the most interesting contributions of this paper is a list of features, inspired on the human visual system, that a motion and structure recovery algorithm should have: i) the model should grow in detail, or precision as more information (frames) becomes available, ii) methods should be robust to non-rigidity, iii) they should be able to deal with short viewing periods, and iv) they should have the possibility to integrate other sources of 3-D information. Also interesting is to observe that many of today's methods attempt to provide some of these features, but not often is it more than one and usually in a very limited way. On the technical contributions part, Ullman introduces a method where rigidity, measured as the preservation of the distance between all pairs of points, is maximized to generate a 3-D model that describes a series of observations – a premise that motivated recent work on non-rigid structure from locally rigid motion [43]. The author suggests a motion segmentation algorithm based on the same principle.

Kanatani [23] is interested in analyzing the capacity of models that explain any type of data, not just trajectories. The importance of carefully selecting a model with the right capacity is explained through a series of simple examples. For instance, whether to fit an ellipse or a line to a cloud of points with linear correlation: a particularly related issue to the model selection problem for simultaneous segmentation of degenerate and non-degenerate motions. The solution is based on the geometric Akaike information criterion (GAIC), which balances the geometric goodness of fit with a penalty for the complexity of the model. Some issues still remain, however, since the GAIC requires knowing the covariance matrix of the noise. But once this matrix estimated (up to scale, at least) the GAIC of different models can be used to identify the best model from those available.

Besides the random-sampling model-instantiation method (see Section 2.2.5 for details) introduced by Torr [48], this paper is also one of the first few to discuss the issue of model selection. In the proposed solution, motion model memberships for each trajectory are estimated via maximization of the Geometric Robust Information Criterion (GRIC).

Finally, Jia and Martinez [22] develop some theory on low rank matrix factorization, particularly with respect to the effects of noise. Their argument is that the estimation of the subspace of a matrix whose vectors are similar

is more sensitive to noise than that of a matrix in which its vectors clearly describe the directions of variation. This is relevant to the problem of motion segmentation as it aligns with the notion of saliency of a motion model.

Observations More than 30 years have passed since Ullman published the seminal paper mentioned above, and the community is still working to achieve Ullman's ideal motion perception algorithm. Interestingly, most papers presented so far try to provide one (and unfortunately only one) of these features, i.e. Schindler [37] models the degree of non-rigidity by estimating the magnitude of the noise, Lakdawalla et al. [27] integrate light and texture as other sources of 3-D information, just to name a couple. Probably, a reasonable way to think of the motion segmentation problem could be on how to integrate Ullman's ideal algorithm features.

The combined capacity of a set of motion models is almost always dictated by effective rank of the observation matrix and often defines the type and number of individual motion models used to explain each set of coherently moving trajectories. At the same time, the mechanisms to compute the necessary capacity of an individual motion model are very related to the problem of noise estimation. Arguably, all these dependencies are bidirectional and typically under-constrained, since one can always model additional tracking noise, for instance, using a richer motion model.

2.3 Applications

The following are just a few examples that use the notion of motion towards a more general purpose: image segmentation, feature tracking and video retrieval. Shi, Belongie, Leung and Malik [39] propose a graph-cut based image or video segmentation method. The similarity metric between spatially and temporally neighboring pixels includes texture, color and motion estimation. Sivic, Schaffalitzky and Zisserman [41] propose grouping of features from objects in video sequences with the goal of automatic video retrieval. In their work, first planar and then epipolar geometries are estimated via RANSAC to automatically generated tracks across neighboring frames. The goal is to merge cohesive sets of tracks, which are then used as features that must be identified to group video sequences with similar image content. Buchanan and Fitzgibbon [7] propose a feature tracking method with priors

based on multi-view geometry. The predicted locations are computed by fitting a local motion model to subsets of tracked features. The prior model is a Gaussian distribution centered at the predicted location. Finally, Furukawa and Ponce [13] propose a model with sophisticated engineering to reconstruct highly detailed 3-D models from 2-D images using photometric and epipolar constraints.

2.4 Benchmark Datasets

Tron and Vidal [49] noticed that almost all papers in the area show test results on different sets of sequences and propose a unifying benchmark dataset of images and their corresponding trajectories called the `Hopkins 155`, which includes degenerate, non-degenerate, independent, partially dependent and articulated motions. Some sequences also have some degree of perspective effects.

Zhou, Tang and Wang [3] introduce the `Collective Motion Database` in a paper that measures crowd collectiveness. The scenes in this dataset are challenging and often non rigid, but they provide a challenging environment for any MS algorithm. The dataset contains only images but trajectories can be estimated using any of the available feature tracking methods.

Both of these (and a few other) datasets or sequences are used to evaluate our work.

3 Method Overview

This chapter is an overview of the main contribution of this thesis: a solution to the problem of orthographic rigid motion segmentation of trajectory data (hereafter referred to as the MS problem), that is computationally efficient as well as competitively accurate. The method works under the assumption that the number of independent motions N is known.

N Number of independent motions in the scene.

3.1 Algorithmic Description of the Problem

The input to our method is a $2F \times P$ matrix of trajectory data $\mathbf{W}$ that represents the x and y image coordinates of P feature points tracked over F frames. The output is a $P \times (N+1)$ binary labelling matrix $\mathbf{L}$ that uses a one-hot encoding to indicate that trajectories belong to one of N inlier motion model classes, or to an outlier class. Note that besides the trajectory data and the number of independent motions, no other information is available to the algorithm (*e.g.*, no image data is available).

$\mathbf{W}$ Matrix of trajectory data.

P Number of trajectories in the sequence.

F Number of frames in the sequence.

$\mathbf{L}$ Binary matrix of motion-segmentation labels using one-hot encoding.

Figure 1.1 (on page 2) shows the inputs and outputs of the algorithm using a few frames from an example sequence. The original images are shown on the top row, the trajectories from the corresponding frames are shown in the middle row. The images at the bottom show the algorithm's output as color-coded trajectories overlaid on a gray-scale version of the original images.

As mentioned in Chapter 2, the solution to the MS problem under ideal conditions (noise- and outlier-free trajectories, completely rigid underlying objects, non-degenerate motions or structures, and orthographic projection) is trivial using the Shape Interaction Matrix-based method [8, 9], but almost

none of these conditions are met in real-world sequences. The proposed algorithm is designed to efficiently motion-segment noisy trajectories under the presence of a large number of outliers and with degenerate motions and/or structures. Rigidity and orthographic projections are both theoretically still necessary conditions, although our experiments show very good segmentation performance for sequences with highly non-rigid deformations and under slight perspective effects (see Chapter 7 for details).

The proposed method can be divided into three stages. A brief outline is provided next, with the goal of providing the reader with an idea of how the algorithm is structured. Reasoning about each sub-problem and some motivation for each of the design choices is left to a more detailed overview, presented in Sections 3.3-3.5. Full algorithmic and mathematical details are available in Chapters 4 to 6.

3.2 Short overview

The first step addresses the problem of instantiating plausible hypotheses of models of motion. To achieve this goal, spatially local subsets of trajectories are used to estimate the parameters of what we call a spatially local model of rigid motion $\mathbf{M}$. Ideally, a good motion model should be capable of explaining the motion of all of the trajectories from a single rigid object. The model instantiation method is based on the Random Sampling and Consensus (RANSAC) paradigm, for robustness to outliers, and incorporates the notion of model selection, to allow for degenerate and non-degenerate motions or structures. Several of these local models are instantiated from many different image regions to build the set $\mathbb{C}$ of candidate motion models.

$\mathbf{M}$ Spatially local mod motion.

$\mathbb{C}$ Set of candidate loc models of motion.

The second step is about finding subsets of N motion hypotheses (from $\mathbb{C}$) that most accurately explain the motion of trajectories from all N classes. A tuple of N motion models constitutes what we call a Model Combination $\mathbb{T}_j = (\mathbf{M}_{j_1}, \mathbf{M}_{j_2}, \ldots, \mathbf{M}_{j_N})$. A list of all candidate model-combinations $\mathbb{S} = (\mathbb{T}_1, \mathbb{T}_2, \ldots)$ is then built, either by random sampling from the candidate set $\mathbb{C}$, or by exhaustive combinatorial listing (of all the $\binom{|\mathbb{C}|}{N}$ possible combinations), potentially rendering a very large set $\mathbb{S}$.

$\mathbb{T}$ Tuple of N motion models, a.k.a. Model combination.

$\mathbb{S}$ Tuple of all candida model combinations.

Each model combination $\mathbb{T}_j \in \mathbb{S}$ is then efficiently evaluated using a function $\mathcal{O}(\mathbb{T}_j)$ that promotes prediction accuracy and penalizes model complexity. The objective function $\mathcal{O}(\mathbb{T}_j)$ implicitly uses a segmentation labeling $\mathbf{L}_j$ to

$\mathbf{L}_j$ Labelling associate model combination $\mathbb{T}_j$.

determine the prediction accuracy from the best model. The labeling $\mathbf{L}_j$ is determined by minimum Euclidean distance from each trajectory to a model $\mathbb{M}_i \in \mathbb{T}_j$.

A small subset of the highest ranking model combinations is then re-ranked, using a second objective function $\mathcal{O}'(\mathbb{T}_j)$ that simultaneously fits a noise model for each class. This time, the labelling $\mathbf{L'}_j$ is determined by maximum likelihood using the estimated noise model. And while computationally more expensive, this approach renders increased segmentation accuracy by modelling trajectories with varying levels of noise per class and per frame.

The third step combines the set of labelings $\{\mathbf{L}'_j\}$ from the best ranked model combinations into a final segmentation result. The parameters of the resulting motion models are recovered last, using the resulting segmentation labels and a Matrix Factorization based method.

The remainder of this chapter provides a more detailed overview of the entire system and further explains how each section of the algorithm allows deviations from a specific subset of the ideal conditions.

Side notes are used every time a new variable is introduced, in order to facilitate future reference. An index of variables and their definitions is available on Page xi of the Thesis's Preamble.

3.3 Stage 1: Spatially-Local Instantiation of Motion Models

The first stage of the algorithm requires instantiating plausible hypotheses of spatially local models of rigid motion ($\mathbf{M}$). The input to this stage is the whole observation matrix of trajectory data ($\mathbf{W}$). A parameter determines how many local motion models (*i.e.*, how many $\mathbf{M}_i$s) must be estimated (between 50 and 100 in our implementation), defining the size of the candidate model set $\mathbb{C}$, which is the output of this stage. The model instantiation algorithm is repeatedly executed to estimate the parameters of each candidate model using small subsets of trajectories from a locally-coherent, spatial neighborhood.

Local coherence has been used in the context of MS in the past, for instance as a regularizer for 2D affine motion [52] or as a prior for motion segmentation [16, 17, 18]. We also believe that there is an increased chance of finding trajectories of the same class within a spatially coherent region. To estimate each model $\mathbf{M} \in \mathbb{C}$, a disk-shaped region is randomly chosen. The location of

this region is sampled from a uniform distribution over the image space and its radius is also sampled from a uniform distribution over a range mostly determined by the size of the image. This disk-shaped region defines the subset of locally coherent trajectories $\hat{\mathbf{W}}$ that will contribute towards the estimation of the model $\mathbf{M}$.

$\hat{\mathbf{W}}$ Subset of spatially l trajectories.

A good motion model should accurately explain the motion of all of the inlier trajectories in $\hat{\mathbf{W}}$, but a better model would also spatially extrapolate well, explaining the motion of the rest of the trajectories that originate from the same (and only the same) underlying rigid object.

3.3.1 Motion Modeling

Motion modeling is a key component to many segmentation algorithms [26, 28, 29, 37, 38, 46, 48, 58] because trajectory labels can be assigned based on model prediction accuracy. And while the understanding of the problem is thorough, and the literature abundant, a reliable general method remains elusive, mostly because of the various difficulties associated to motion-encoding data: noise (structured or not), outliers, motion degeneracy, motion dependency, etc.

Our method assumes an affine camera model[1], allowing for orthographic, weak perspective or para-perspective projection, so the parameters of a motion model could be straightforwardly estimated using linear least squares (like matrix factorization [45], Equation 2.5), from the subset $\hat{\mathbf{W}}$. However, despite limiting this subset to a spatially-local neighborhood, trajectories in $\hat{\mathbf{W}}$ may still originate from more than one rigidly moving object, which suggests that the method must be robust to a potentially large number of outliers. The presence of outliers rules out the direct application of least squares techniques to $\hat{\mathbf{W}}$.

Traditionally, the Random Sampling and Consensus (RANSAC) algorithm [12] has been a popular choice for robust estimation of the parameters of models of motion [29, 46, 47, 48, 58, 59], but the standard RANSAC has two important drawbacks in the context of MS. The first one is that it is limited to a single model type –one can estimate the parameters of either degenerate or non-degenerate motions– and the second one is that it requires a criterion to do inlier detection.

[1] an independent affine projection is used per frame and per class to facilitate estimation. The risk is potential model over-fitting which translates in reduced MS accuracy.

The issue of multiple model types has been dealt with in the past [37, 38, 48]. These methods instantiate both degenerate and non-degenerate motion models and leave the problem of model selection [23] to a later stage. An exception is [42], where a non-degenerate (3D Affine) model is initialized using the result of fitting a degenerate (2D Affine) one. Then the 3D Affine model is further optimized to maximize prediction accuracy of trajectory data. The assumption is that, if in fact the object (or the motion) is degenerate, the parameters of the degenerate model will not change after upgrading to a non-degenerate one and optimizing. Unfortunately, it is mathematically possible (and common in practice) for a non-degenerate motion model to describe the motion of two articulated motions [59], in which case the aforementioned method may result in under-segmentation.

Our solution is a modified version of RANSAC that fits a 2D Affine model using 3 trajectories [19] and then analyzes the residual distribution, from the differences between the observations and the model predictions. When the trajectories in $\hat{\mathbf{W}}$ do in fact, originate from a degenerate motion, the model is appropriate and the residual distribution is similar to the distribution of the noise in the data (which we assume isotropic and of small magnitude). If the motion is non-degenerate, the planar model is insufficient and the unaccounted relative depth will manifest as a residual distribution that lies along a line [20], the Epipolar Line. In this case, the parameters needed to upgrade the degenerate model into a non-degenerate one can be directly estimated from the residual distribution, making the process computationally very efficient.

Regarding inlier detection, our criterion is based on the magnitude of the residuals, defined as the distance between the 2D model prediction and the observation for the planar model, or as the shortest distance from the estimated epipolar line (segment) to the observation, for the 3D model. Using this definition, model predictions become model inliers if their residual magnitude is smaller than a dynamically defined threshold. The threshold is determined as the start of the first residual-magnitude gap that is significantly larger than all other residual-magnitude gaps. See Figure 3.1 for a real-data example. This criterion is applied at every frame, and a trajectory becomes an inlier if its predictions satisfy the inlier criterion in at least half of the frames in the sequence.

The inlier detection criterion described above makes two assumptions.

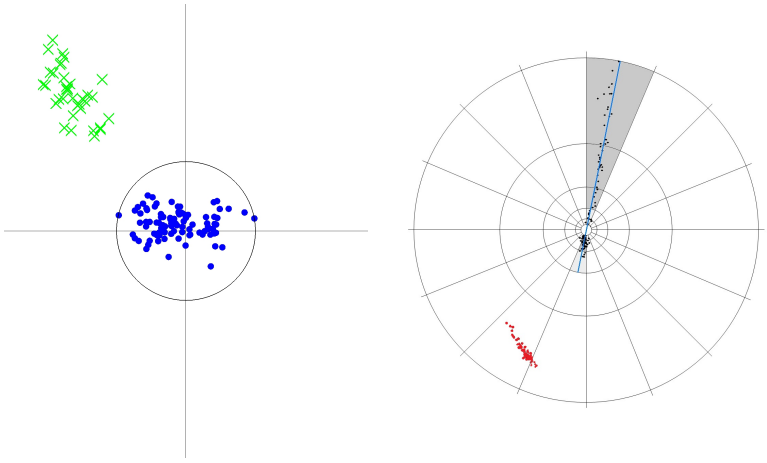


Figure 3.1: Inlier Detection. Plots show one frame of ground-truth color-coded residuals (the result of subtracting the 2D affine model predictions from the observations). The inlier detection criterion is based on magnitude gaps between residuals. Left: inlier residuals from a degenerate motion (blue dots) and outliers (green 'x's). The automatically determined inlier threshold is shown as a black circle. Right: inlier residuals from a non-degenerate motion (black) and outliers (red). The automatically fitted epipolar line segment is shown in light blue. Red residuals are too far from the line segment.

First, that the magnitude of trajectory noise must be smaller than the magnitude of the motion differential between independently moving objects, and second, that a salient magnitude gap can be reliably identified. The first assumption is very often true, as it lies close to the very definition of motion independence of corresponding points between pairs of images [22]. The second assumption is more frequently violated, since it is possible that large gaps may occur by pure serendipity, creating spurious gaps, or gaps may disappear due to noisy trajectories. Still the consequences of casual gap-finding difficulties are ameliorated by the integration of the criterion across all frames in the sequence.

Furthermore, it must also be kept in mind that the motion instantiation algorithm is run many times to create a pool of motion models ($\mathbb{C}$), from which only the best few subsets of size N are selected to do the final segmentation result, as explained next.

3.4 Stage 2: Model Selection

The second stage of the algorithm is about generating model combinations $\mathbb{T}$ and then ranking them according to a metric that prefers combinations that better model the data and that do so more efficiently. The idea is not new [23, 29, 37, 38], but most of the existing methods do so greedily, mainly for computational efficiency, and suffer the consequences of potential loss of accuracy.

In our method, model combinations ($\mathbb{T}$) are generated either by random sampling from the candidate motions set $\mathbb{C}$ or by exhaustively listing all possible $\binom{|\mathbb{C}|}{N}$ combinations of size N from $\mathbb{C}$. We define $\mathbb{S} = \{\mathbb{T}_1, \mathbb{T}_2, \ldots\}$ as the set that contains all the model combinations that will be considered.

Each model combination $\mathbb{T}_i \in \mathbb{S}$ is then ranked according to a metric that evaluates how good the models in $\mathbb{T}_i = \{\mathbf{M}_{i_1}, \mathbf{M}_{i_2}, \ldots, \mathbf{M}_{i_N}\}$ are at characterizing all of the trajectory data for the N independently moving objects of the scene. An outlier model [48] is also available to capture trajectories that cannot be explained by any of the models in $\mathbb{T}_i$. The evaluation metric promotes prediction accuracy while penalizing model complexity, modeling redundancy and excessive use of the outlier class.

The input to this stage is the set of candidate models $\mathbb{C}$, as well as the parameter that determines the number of model combinations to draw (limited above by $C^N_{|\mathbf{C}|}$), implicitly determining the size of $\mathbb{S}$. The output is a subset from $\mathbb{S}$ that includes the highest ranking model combinations, as well as the associated evaluation scores.

Evaluating the overall prediction accuracy of a given model combination $\mathbb{T}_i$ requires determining which model predicts each of the trajectories. The trivial solution to this problem consists of assigning the label that corresponds to the model $\mathbf{M}_j \in \mathbb{T}$ that produces the smallest Euclidean residual, for each trajectory (and without any consideration of spatial coherence).

However, while finding minimum Euclidean distances is computationally inexpensive, using this type of metric requires assuming that the underlying trajectory noise has an isotropic distribution of equal magnitude across all independent motions. In practice, this assumption is often violated, typically due to deviations from the orthographic projection (perspective effects), non-rigidness, or other uncounted image formation effects (like radial distortion).

An alternative way of dealing with this problem consists of estimating the

parameters of a joint model that better captures the distribution of the noise for each motion at each frame, like a 2D Normal distribution, and then assign labels using the model with maximum likelihood. Certainly, this approach accounts for noise distributions of different magnitudes, and captures some of the structure of the distribution, but requires running an iterative algorithm (Expectation-Maximization) to estimate the noise model's parameters for each motion model and for each frame, making it computationally expensive and potentially unfeasible, particularly when $|S|$ is large.

Our solution to this problem is a combination of the two approaches described above. The Euclidean metric based approach is used to efficiently prune the set S, keeping only a small subset of the best scoring model combinations. The surviving combinations are then scored using the Gaussian Mixture Model (GMM) metric to get a more accurate ranking of the combinations that are making the better motion predictions.

3.5 Stage 3: Model Averaging

The third stage of the algorithm merges the labels from the best scoring model combinations into a final segmentation result. The input to this stage is the subset of highest scoring model combinations and their associated scores. The outputs are the final segmentation labels as well as the resulting models of motion.

While it is possible to output the labeling from the best ranked model combination, according to the GMM evaluation metric, as the final segmentation result, we found that by incorporating the labellings from some of the highest ranked model combinations, the resulting segmentation accuracy could be increased, and the labeling variability, due to the random nature of the algorithm, decreased.

The key observation is that the segmentation results from many of the top ranking model combinations are similar, and mostly correct except for a small number of trajectories. This redundancy of correct labellings can be explained because the Single Model Instantiation stage (briefly introduced in Section 3.3) typically produces several good models for each of the motions in the scene. This leads to multiple model combinations that can correctly explain the motion of most of the trajectories. The problem is that because all of the single models of motion are instantiated with a spatially limited subset of

trajectories, some of them need to extrapolate the motion of trajectories that lies much further from the region where the control trajectories live, leading to occasional mis-classification errors.

We propose a weighted average as a way to incorporate the labelings of multiple segmentation results. The weights are a function of the evaluation metric score, giving more weight to labellings that result from better scoring model combinations.

The method builds an affinity matrix [10, 55, 59, 60] for all of the trajectories in the observation matrix $\mathbf{W}$. The $z_{(i,j)}$ entry of the affinity matrix $\mathbf{Z}$ corresponds to a weighted average that indicates how often trajectories i and j given the same label. The resulting affinity matrix is clustered using a standard spectral clustering technique. The resulting labels from this algorithm represent the final segmentation result.

$\mathbf{Z}$ Affinity matrix of trajectory labels.

The Matrix Factorization method is finally used to compute estimates of the parameters of the final motion models, using on all of the trajectories with the same label for each of the N possible (inlier) classes.

4 Local Motion Model Fitting

This chapter introduces the Local Motion Model Fitting (LMMF) algorithm, a method to robustly estimate both the capacity and the parameters of a motion model ($\mathbf{M}$). This model characterizes the motion of a spatially local subset of trajectories from a single, rigidly moving object, when imaged under orthography, and in the presence of noise and outliers. The novelty of the approach lies in the combination of a robust, random sampling-based, parameter estimation algorithm, coupled with a representation of motion that allows us to seamlessly upgrade a degenerate model of motion into a non-degenerate one (see Section 4.2.2), and does the upgrade at a fraction of the computational cost of estimating it from scratch. This results in an efficient model instantiation method that considers both types of models without sacrificing robustness. Another key component to the LMMF algorithm is a novel inlier detection mechanism that uses the distribution of the residuals (the differences between the model predictions and the actual data observations) to automatically determine an inlier criterion that adapts to the magnitude of the underlying noise (see Section 4.2.4).

The LMMF algorithm lies at the core of the MS pipeline, since the set of candidate models of motion $\mathbb{C} = \{\mathbf{M}_1, \mathbf{M}_2, \ldots\}$ is populated by repeatedly running this algorithm with different locally coherent subsets of input trajectories (see Chapter 5 for details).

4.1 Overview of the Method

The LMMF is based on the Random Sampling and Consensus (RANSAC) parameter estimation paradigm [12]. The choice is certainly motivated by the characteristic robustness to outliers and computational efficiency of RANSAC, despite the lack of optimality guarantees regarding the accuracy of the resulting model's predictions, or the size of the data subset that is actually modeled. But in addition to speed and robustness, the choice of RANSAC is also motivated by two problem-specific reasons. The first one is the ability to leverage the notion of spatially local support, in order to observe the assumption of spatial coherence for trajectories originating from the same object. The second one is the ability to optimize an objective function that must operate with different model types and that includes a term that penalizes model complexity.

Each RANSAC iteration of the LMMF algorithm has at least two, and possibly three stages, depending on whether the underlying trajectory data is degenerate or not. During the first stage, a 2D Affine model is estimated and inliers to this model are found. In the second stage, the residuals of the 2D Affine model are analyzed to determine whether the underlying data originates from a degenerate or a non-degenerate motion. When the motion is found to be non-degenerate, the third stage determines the parameters of the 3D model and finds the corresponding inlier subset.

In: Locally-Coherent Trajectories, Out: A Model of Motion

The input to the LMMF algorithm is a spatially-local subset of trajectory data $\hat{\mathbf{W}}_{[2F\times I]} \subseteq \mathbf{W}_{[2F\times P]}$. The use of a spatially-coherent neighborhood increases the likelihood of choosing control trajectories from the same object, exponentially decreasing the computational cost of finding an uncontaminated set of control points (see Section 4.1.1 for details). Figure 4.1 shows two examples of the spatial locations of trajectories in $\hat{\mathbf{W}}$ (white dots), for two different regions (the regions within the white circles), at arbitrary frames. Note that because the location and size of the support region are randomly determined (see Section 5.1 for details), the support region is oblivious to the underlying trajectory classes, and consequently $\hat{\mathbf{W}}$ often includes trajectories from more than one independently moving object.

$\hat{\mathbf{W}}$ Spatially-local subs trajectory data.

I Number of trajector $\hat{\mathbf{W}}$.

The ideal output of the LMMF algorithm is the model $\mathbf{M}$ that predicts

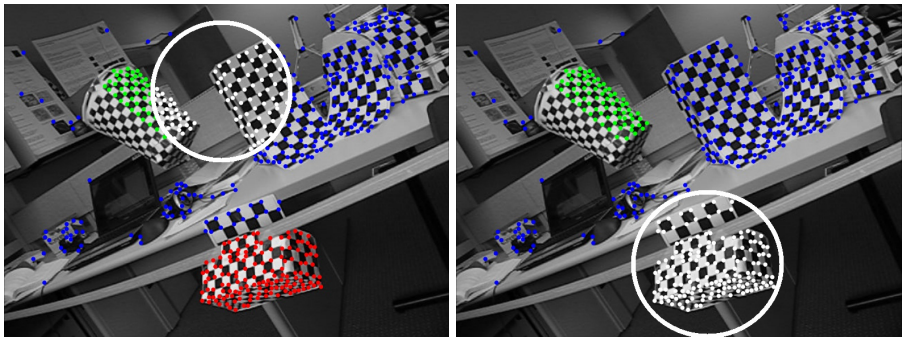


Figure 4.1: Spatially-local subsets of trajectories $\hat{\mathbf{W}}$ (white markers). These two examples are typical inputs to the LMMF algorithm. Note that in both cases, $\hat{\mathbf{W}}$ contains trajectories from multiple independently moving objects.

the motion of the most salient subset of rigidly moving trajectories within $\hat{\mathbf{W}}$, with the best accuracy and with the simplest model possible.

4.1.1 Random Sampling

As is typical for RANSAC, the output model $\mathbf{M}$ is the one that optimizes an objective function

$$\mathbf{M} = \underset{\{\mathbf{M}_i\}_{i=1}^{K}}{\operatorname{argmin}} \mathcal{O}(\hat{\mathbf{W}}, \mathbf{M}_i) \tag{4.1}$$

K Number of RANSAC trials.

over a (limited) set of K model candidates. The objective function $\mathcal{O}(\hat{\mathbf{W}}, \mathbf{M})$ promotes prediction accuracy while penalizing model complexity and modeling accuracy (details in Section 4.3).

Number of RANSAC Trials (K)

The probability of choosing an uncontaminated set of 3 control trajectories, necessary to compute a 2D Affine motion model, from a dataset with a ratio λ_r of same-class trajectories, after K trials is

λ_r Expected ratio of same-class trajectories within $\hat{\mathbf{W}}$.

$$p = 1 - (1 - \lambda_r^3)^K. \tag{4.2}$$

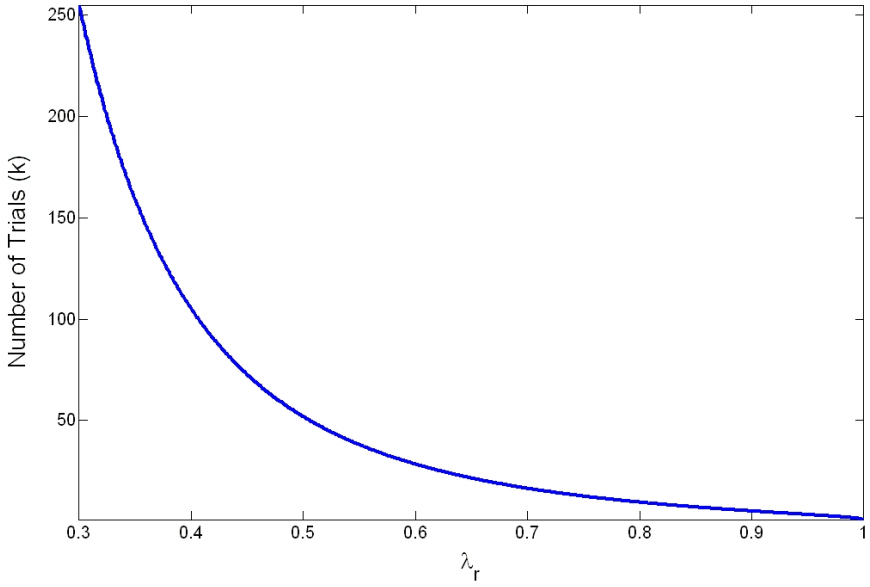


Figure 4.2: Number of Trials (K) necessary to find an uncontaminated subset of 3 trajectories with an inlier ratio of $\lambda_r \in [0.3, 0.999]$ with probability $p = 0.999$.

This implies that the number of trials needed to find a subset of 3 same-class trajectories with probability p is

$$K \geq \frac{\log(1-p)}{\log(1-\lambda_r^3)}. \tag{4.3}$$

A common assumption is that trajectories from the same underlying motion are locally coherent. Hence, a compact region is likely to increase the ratio λ_r, exponentially reducing K, and with it, RANSAC's computation time by a proportional amount. Figure 4.2 shows a plot of the number of trials (K) necessary to find an uncontaminated subset of 3 trajectories with an inlier ratio of $\lambda_r \in [0.3, 0.999]$, with probability $p = 0.999$. Note the exponential decay of k as λ_r grows. In our implementation we use $K \in [50, 100]$ RANSAC trials per instantiated motion model $\mathbf{M}$, typically $K = 70$.

From Control Trajectories to a Model of Motion

Each candidate model $\mathbf{M}_i$ from Equation 4.1 is constructed using the deterministic mapping

$$(\hat{\mathbf{W}}, \mathbb{D}_i) \rightarrow \mathbf{M}_i \tag{4.4}$$

which takes a spatially local subset of trajectory data $\hat{\mathbf{W}}$, as well as a set of three control indices $\mathbb{D}_i = \{d_p, d_q, d_r\}$ to produce the model $\mathbf{M}_i$ (details in Section 4.2).

$\mathbb{D}$ Set of 3 control indices.

Control indices are randomly sampled from the set of increasing triplets:

$$\mathbb{D}_i \in \{(p,q,r) : p,q,r \in \{1,\ldots,I\},\ p < q < r\} \tag{4.5}$$

with uniform distribution. These indices determine the control trajectories $\{\mathbf{w}_{d_p}, \mathbf{w}_{d_q}, \mathbf{w}_{d_r}\} \subset \hat{\mathbf{W}}$ that ultimately determine the parameters of $\mathbf{M}$ for each of the K RANSAC trials (as explained in Sections 4.2.1 to 4.2.5).

4.2 The Parameters of a Motion Model

A motion model $\mathbf{M}$ is comprised by a 5-tuple of parameters:

$$\mathbf{M} = (\mathbb{A}, \mathbb{E}, \mathbf{B}, \sigma, \omega)\,. \tag{4.6}$$

Assuming (for the sake of clarity) that the model $\mathbf{M}$ explains the motion of all the trajectories in $\hat{\mathbf{W}}$, the parameters of $\mathbf{M}$ can be explained as follows.

The tuple

$$\mathbb{A} = (\mathbf{A}^1, \mathbf{A}^2, \ldots, \mathbf{A}^F) \tag{4.7}$$

$\mathbb{A}$ Tuple of F 2D Affine Transformations.

with

$$\mathbf{A}^f \in \begin{bmatrix} \mathbf{P}^f & \mathbf{q}^f \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{4.8}$$

contains the 2D Affine transformations $\mathbf{A}^f$ that project trajectories in $\hat{\mathbf{W}}$ from an arbitrary base frame b onto the f^{th} frame (with $\mathbf{A}^b$ the identity matrix). Each 2D Affine projection $\mathbf{A}^f$ is composed by a non-singular 2×2 matrix $\mathbf{P}^f$ and a column vector $\mathbf{q}^f \in \mathbb{R}^2$, as indicated in Equation 4.7. When the underlying motion of the trajectories in $\hat{\mathbf{W}}$ is in fact degenerate, the 2D Affine projections ($\mathbf{A}^f$) are sufficient to explain their motion, except for perturbations due to noise, hence defining an appropriate evaluation metric for verification

$\mathbf{A}^f$ 2D Affine projection matrix for frame f.

b Arbitrary base frame.

of coherent rigid motion.

On the contrary, when the underlying object is non-planar, and its motion non-degenerate, the predictions made by a planar model will fail to account for the effects of relative depth. In this case, rigid motion coherence is validated using the 2D Affine Plus Epipolar constraint (to be explained in Section 4.2.2), which, in addition to the 2D Affine projections, also requires estimating an Epipolar Direction vector for every frame $\mathbf{e}^f$.

$\mathbf{e}^f$ Epipolar direction a frame f.

The tuple of all (F) Epipolar Direction vectors is:

$\mathbb{E}$ Tuple of F Epipolar Direction Vectors.

$$\mathbb{E} = (\mathbf{e}^1, \mathbf{e}^2, \ldots, \mathbf{e}^F) \tag{4.9}$$

with $\mathbf{e}^f \in \mathbb{R}^2$ and $\mathbf{e}^b = [0,0]^\top$.

Now, assume that the model $\mathbf{M}$ only explains the motion of a subset of the trajectories in $\hat{\mathbf{W}}$. The rest of the trajectories could be from an independently moving object, from tracking errors, from self occluding trajectories or from any other outlier source. In this (more common) case, the matrix $\mathbf{B} \in \{0,1\}^{[F \times I]}$ indicates whether the model $\mathbf{M}$ is making sufficiently accurate predictions for trajectory $\mathbf{w}_i$, at frame f (inlier, $b_i^f = 1$) or not (outlier, $b_i^f = 0$).

$\mathbf{B}$ Binary matrix of in trajectories per frame.

The tuple

$\boldsymbol{\sigma}$ Tuple of estimates magnitude of the noise

$$\boldsymbol{\sigma} = (\sigma^1, \sigma^2, \ldots, \sigma^F) \tag{4.10}$$

with $\sigma \in \mathbb{R}^+$, contains estimates of the magnitude of the noise at each frame.

And finally, the variable $\omega \in \{2\text{D}, 3\text{D}\}$ indicates whether the model is explaining a subset of trajectories with either degenerate or non-degenerate motion.

ω Model capacity ind variable.

Details regarding the computation of each of these model parameters ($\mathbb{A}$, $\mathbb{E}, \mathbf{B}, \boldsymbol{\sigma}, \omega$) come next.

4.2.1 2D Affine Transformations ($\mathbb{A}$)

The 2D Affine transformation that projects the location of a set of trajectories indexed by $\mathbb{D}_i = (p, q, r)$, from an arbitrary base frame b to a target frame f can be computed using

$$\mathbf{A}^{b \to f} = \begin{bmatrix} \mathbf{w}_p^f & \mathbf{w}_q^f & \mathbf{w}_r^f \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{w}_p^b & \mathbf{w}_q^b & \mathbf{w}_r^b \\ 1 & 1 & 1 \end{bmatrix}^{-1}, \tag{4.11}$$

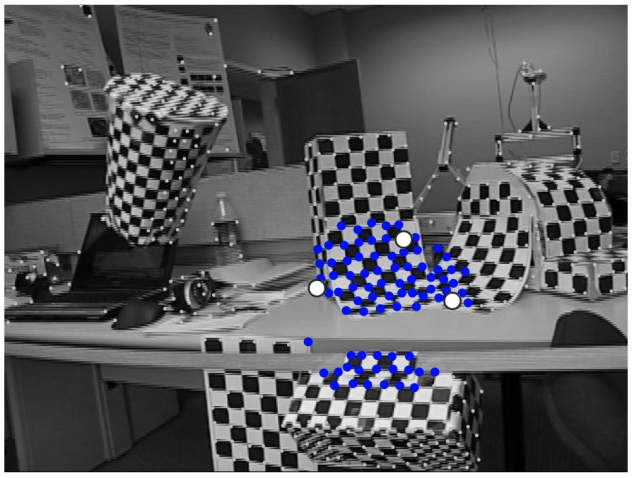


Figure 4.3: Control trajectories from an arbitrary base frame, $\mathbf{w}_p^b$, $\mathbf{w}_q^b$ and $\mathbf{w}_r^b$ (white dots) randomly chosen from within the set $\hat{\mathbf{W}}$ of locally coherent trajectories (blue dots). The rest of the trajectories are shown in light gray.

where $\mathbf{w}_i^f = [w_{i,x}^f, w_{i,y}^f]^\top$ corresponds to the x and y coordinates of the i^{th} trajectory at frame f. The inverse of the rightmost matrix of Equation 4.11 exists so long as the control points $\mathbf{w}_p^b$, $\mathbf{w}_q^b$ and $\mathbf{w}_r^b$ are not collinear. For this reason, before attempting to estimate these 2D Affine transformations, a function determines whether the triangle defined by the points in $\{\mathbf{w}_p^b, \mathbf{w}_q^b, \mathbf{w}_r^b\}$ has a length 10 times bigger than the width, or more. If this is the case, the control triplet $\mathbb{D}_i$ is discarded and a new one is drawn (randomly, with uniform distribution from the set defined in Equation 4.5), and the process repeated until a triplet passes the condition.

For simplicity $\mathbf{A}^{b\to f}$ is referred to as $\mathbf{A}^f$ (consequently $\mathbf{A}^b$ is the identity matrix). Figure 4.3 shows an example of a randomly drawn set of control trajectories (in white), from within a spatially coherent set $\hat{\mathbf{W}}$ (in blue).

4.2.2 Model Capacity

This section explains how the proposed method determines the most suitable motion model type (degenerate or non-degenerate) that must be used to evaluate motion coherence for the majority of the data in $\hat{\mathbf{W}}$.

Assuming the trajectory data $\hat{\mathbf{W}}$ is contaminated with isotropic noise of small magnitude, when the underlying object is planar, or if its motion (with

respect to the camera) is degenerate, the 2D Affine model is sufficient to explain the motion of its trajectories. In this scenario, the only unaccounted-for displacements are due to noise. We therefore assume the residuals (the differences between the predictions made by the model, and the observations) will also have an isotropic distribution of small magnitude.

On the contrary, when the underlying structure is non-planar and its motion non-degenerate, the residuals from the 2D Affine model will show that the effects of relative depth remain to be accounted for.

One of the key contributions of the LMMF algorithm that we propose is the use of an alternative metric to evaluate rigid motion coherence of a subset of non-degenerately moving trajectories, instead of the traditional 3D Affine model. The metric uses the 2D Affine plus Epipolar (2DAPE) decomposition (explained below), which only requires fitting a 2D Affine model (which is estimated anyway, to test the planar motion hypothesis), as well as an estimate of the Epipolar direction $\mathbf{e}^f$, but without needing an estimate of relative depth for each trajectory.

2D Affine Plus Epipolar (2DAPE) Decomposition

The description of the 2DAPE decomposition begins with some notation. The camera coordinates $\mathbf{x}_c^f \in \mathbf{R}^3$ with respect to the f^{th} camera can be derived from the world coordinates of a point $\mathbf{x}_w = [x_w, y_w, z_w]^\top$ using:

$$\begin{bmatrix} \mathbf{x}_c^f \\ 1 \end{bmatrix} = \mathbf{C}_E^f \begin{bmatrix} \mathbf{x}_w \\ 1 \end{bmatrix} \tag{4.12}$$

where the Extrinsic Calibration Matrix $\mathbf{C}_E^f$ is of the form

$$\mathbf{C}_E^f = \begin{bmatrix} \mathbf{R}^f & -\mathbf{R}^f\mathbf{t}^f \\ \mathbf{0} & 1 \end{bmatrix} \tag{4.13}$$

with $\mathbf{R}^f$ a 3×3 rotation matrix, and $\mathbf{t}^f \in \mathbb{R}^3$ is the location of the center of projection of the f^{th} camera in world coordinates.

The mapping to image coordinates ($\mathbf{x}_f$) using an orthographic projection can be done using:

$$\begin{bmatrix} \mathbf{x}^f \\ 1 \end{bmatrix} = \mathbf{C}_I\mathbf{C}_E^f \begin{bmatrix} \mathbf{x}_w \\ 1 \end{bmatrix}, \tag{4.14}$$

where the matrix $\mathbf{C}_I$ corresponds to:

$$\mathbf{C}_I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.15}$$

which simply discards the depth coordinate (*i.e.,* the distance from the center of projection to the point $\mathbf{x}_w$ in the viewing direction).

In the context of motion segmentation, what is necessary is a constraint that verifies rigid motion coherence of trajectories from one frame to another (*i.e.,* from a base frame b to an arbitrary frame f).

Now, assuming we had the camera coordinates for all the trajectories at the base frame, it would be possible to estimate the image coordinates at frame f using:

$$\begin{bmatrix} \mathbf{x}^f \\ 1 \end{bmatrix} = \mathbf{A}_{3D}^{b \to f} \begin{bmatrix} \mathbf{x}_c^b \\ 1 \end{bmatrix}. \tag{4.16}$$

with

$$\mathbf{A}_{3D}^{b \to f} = \mathbf{C}_I \mathbf{C}_E^f (\mathbf{C}_E^b)^{-1}. \tag{4.17}$$

Camera coordinates, however, are not available. The only available observations are image coordinates, but Equation 4.14 shows that in fact, if image coordinates are available, the only missing component to get camera coordinates is the point's relative depth, so we could write Equation 4.16 as a function of image coordinates ($\mathbf{x}^f$), and assume we could use an estimate of relative depth δz, as in:

$$\begin{bmatrix} \mathbf{x}^f \\ 1 \end{bmatrix} = \mathbf{A}_{3D}^f \begin{bmatrix} \mathbf{x}^b \\ \delta z \\ 1 \end{bmatrix}, \tag{4.18}$$

with $\mathbf{A}_{3D}^f$ an affine matrix of the form

$$\mathbf{A}_{3D}^f = \begin{bmatrix} \mathbf{P}^f & \mathbf{e}^f & \mathbf{q}^f \\ \mathbf{0} & 0 & 1 \end{bmatrix}. \tag{4.19}$$

From 4.18 and 4.19, it is clear that the image coordinates of $\mathbf{x}^f$ can be written

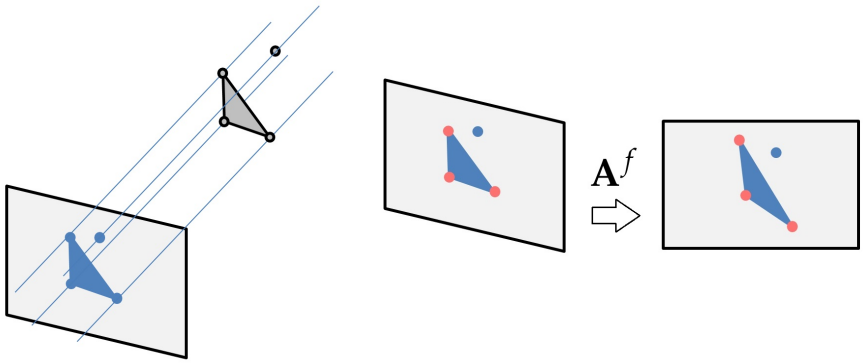


Figure 4.4: Left: The orthographic projection of four points onto the image plane. Right: The result of using a 2D Affine Projection $\mathbf{A}^f$ to estimate the locations of the corresponding points onto the f^{th} frame.

as the combination of a 2D Affine Projection ($\mathbf{A}^f$ from Equation 4.7) plus the Epipolar displacements (along $\mathbf{e}^f$ from Equation 4.9), due to relative depth, as in:

$$\begin{bmatrix} \mathbf{x}^f \\ 1 \end{bmatrix} = \mathbf{A}^f \begin{bmatrix} \mathbf{x}^b \\ 1 \end{bmatrix} + \delta z \begin{bmatrix} \mathbf{e}^f \\ 0 \end{bmatrix} \tag{4.20}$$

Geometrically, the left side of Figure 4.4 shows the projection of four 3D points onto an image plane. Assume that it is the base frame. The right side of the same figure shows the effect of using a 2D Affine transformation to estimate the position of the corresponding points onto an arbitrary frame f. The three pink points were used as control to estimate the parameters of $\mathbf{A}^f$, using Equation 4.11.

Figure 4.5 shows the actual observations of the same four points, as pink circles. The co-planar control trajectories (the ones that form the triangle) are perfectly projected by the 2D Affine model, but the off-plane trajectory results in mis-estimation (highlighted by the red line) in the direction parallel to the epipolar direction (green line).

Figure 4.6 shows a similar plot, but this time additional off-plane points are included. The effect of applying the 2D Affine transformation is shown in blue (mostly a compression along the horizontal direction as well as an elongation on the vertical direction). The real observations of the new points are shown in pink. Note how all of the mis-estimations, positive or negative

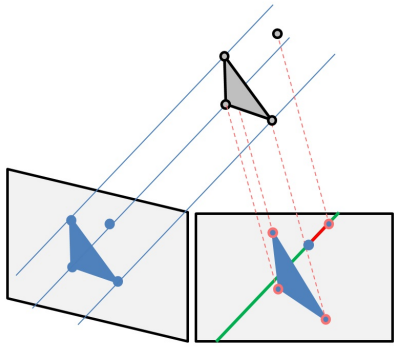


Figure 4.5: Predictions using the 2D Affine model $\mathbf{A}^f$, compared to the actual observations. Points that are co-planar to the control are correctly modeled, but off-plane trajectories result in mis-estimations along the direction $\mathbf{e}^f$ of the epipolar line (in green)

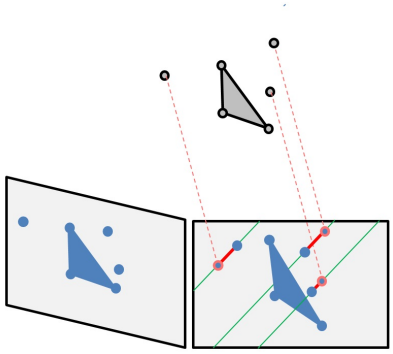


Figure 4.6: More predictions of off-plane points to show that the residuals are all oriented along the epipolar line ($\mathbf{e}^f$), albeit with different magnitudes.

(shown as red lines) lie along the green lines, parallel to the epipolar line. We refer to these mis-estimations as residuals, formally defined next.

2D Affine Model Residuals

$\mathbf{r}_i^f$ Residual for trajectory i at frame f.

The 2D Affine Model residual of the i^{th} trajectory at the f^{th} frame ($\mathbf{r}_i^f$) is defined as the difference between the model prediction, estimated as the projection of the trajectory at a base frame ($\hat{\mathbf{w}}_i^b$) onto the f^{th} frame, and the

actual data observation $\hat{\mathbf{w}}_i^f$:

$$\begin{bmatrix} \mathbf{r}_i^f \\ \vec{0} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{w}}_i^f \\ \vec{1} \end{bmatrix} - \mathbf{A}^f \begin{bmatrix} \hat{\mathbf{w}}_i^b \\ \vec{1} \end{bmatrix}, \tag{4.21}$$

where $\mathbf{r}_i^f$, $\hat{\mathbf{w}}_i^b$ and $\hat{\mathbf{w}}_i^f$, are all 2D vectors (image coordinates).

Figure 4.7 shows 2D Affine model residuals for both a degenerate as well as a non-degenerate motion from real data. Figure 4.7a shows the original trajectory data at an arbitrary frame, overlaid on the original image. Figure 4.7b shows the 2D Affine residuals of a set of non-planar trajectories undergoing a non-degenerate motion. Because the effect of relative depth remains unaccounted for, the residuals distribute along the epipolar line $\mathbf{e}^f$. In contrast, Figure 4.7c shows the residuals of a degenerate motion, where the distribution is similar to that of the underlying tracking noise (isotropic and of small magnitude). In this particular case, the motion is degenerate because the camera only rotates along the center of projection.

The characteristic line distribution of the residuals of the non-degenerate case will be used to determine model type, as explained next.

Determining Model Capacity from Residual Data (ω)

We use the linear distribution of residuals along a single direction as an indicator for the presence of a rigid non-planar object undergoing non-degenerate motion (as in Figure 4.7b). The covariance of the residuals helps determine if this is the case. For this purpose, let

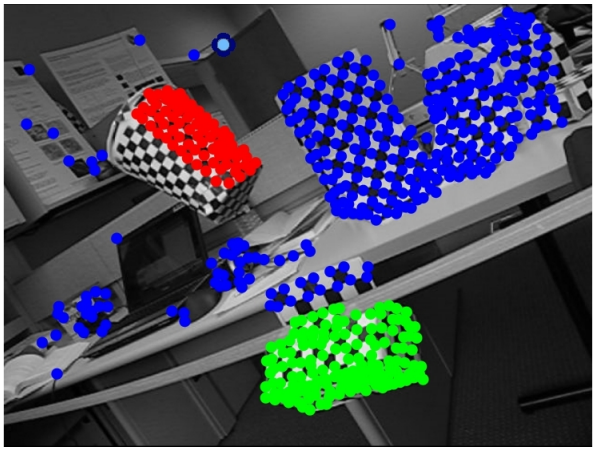
$$\hat{\mathbf{R}}^f = \begin{bmatrix} \mathbf{r}_1^f & \mathbf{r}_2^f & \dots & \mathbf{r}_I^f \end{bmatrix}^\top \tag{4.22}$$

be the matrix (with $\hat{\mathbf{R}}^f \in \mathbb{R}^{I\times 2}$) that contains the 2D Affine model residuals (estimated using Equation 4.21) of all the trajectories in $\hat{\mathbf{W}}$.

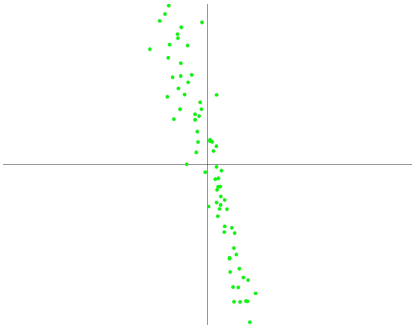
The Singular Value Decomposition (SVD) of the covariance of $\hat{\mathbf{R}}^f$ can then be written as

$$\mathbf{U}\mathbf{S}\mathbf{V}^\top = \mathtt{svd}\left(\frac{1}{I}(\hat{\mathbf{R}}^f)^\top \hat{\mathbf{R}}^f\right), \tag{4.23}$$

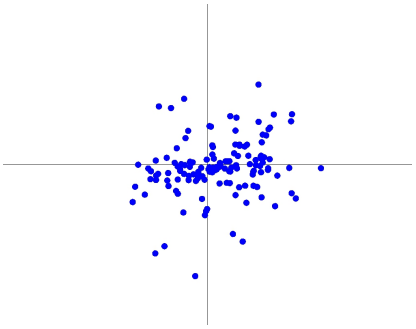
with singular values denoted as $\mathbf{S} = (s_1, s_2)$ with $s_1 \geq s_2$. When the residuals are distributed along a line, the residuals exhibit a large ratio between the largest and the smallest singular value $\frac{s_1}{s_2} \gg 1$. On the contrary, when



(a) Ground truth, color coded trajectories at frame f.



(b) Non-Degenerate motion residuals from the object in (a) with green trajectories.



(c) Degenerate motion residuals from a subset of blue trajectories from (a). This motion is degenerate because the camera only rotates.

Figure 4.7: Residual distributions for a degenerate and a non-degenerate motions.

the degenerate model is sufficient to explain the motion of trajectories, the only unaccounted mis-estimations are due to noise. Because the noise is assumed isotropic, the covariance of the residuals should also be isotropic, and consequently $\frac{s_1}{s_1} \approx 1$ should hold.

The ratio of largest over smallest singular values was empirically validated and found to be a good indicator of when upgrading to a 3D model is

necessary, and the estimate of model capacity (ω) is formally defined as:

$$\omega = \begin{cases} \text{3D}, & \frac{s_1}{s_2} > 1 + \lambda_\omega \\ \text{2D}, & \text{otherwise.} \end{cases} \tag{4.24}$$

In our implementation, the value of λ_ω is 1.3.

4.2.3 Non-Degenerate Coherent Motion

When sufficient evidence of $\omega = \text{3D}$ is found, the 2D Affine motion model must be upgraded to allow verification of non-degenerate coherent motion for the trajectories in $\hat{\mathbf{W}}$. We propose using the 2DAPE decomposition (Section 4.2.2), which indicates that a 2D Affine model can be upgraded to model the motion of non-planar objects by incorporating an Epipolar Direction $\mathbf{e}^f$, together with estimates of the relative depth δz at the base frame for each trajectory, rendering the 3D equivalent of the 2D Affine model.

However, instead of using a full 3D Affine model, we found that non-degenerate motion coherence could be validated simply by checking for proximity of the residuals to the Epipolar line. This means that while it will not be necessary to find estimates of relative depth δz for each trajectory, the estimate of the Epipolar direction $\mathbf{e}^f$ is still necessary.

Notice that the presence of outliers in $\hat{\mathbf{W}}$ leads to outliers in $\mathbf{R}^f$ (residuals that will not generally align with $\mathbf{e}^f$), which means that the estimate of $\mathbf{e}^f$ must be done robustly, as explained next.

Estimating the Epipolar Direction ($\mathbf{e}^f$)

If the trajectories in $\hat{\mathbf{W}}$ were outlier free, the estimate of $\mathbf{e}^f$ would be trivially available from the SVD of the residual covariance (Equation 4.23). In this ideal case the epipolar direction would simply be the right singular vector associated to the singular value of largest magnitude ($\mathbf{e}^f = \mathbf{v}_1$, assuming $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2]$). But because of the potentially large leverage that outliers have on the estimate of principal directions using SVD, a more robust method is used instead.

Robustness to outliers is achieved by using a mixture of different techniques. The first one consists of fitting a line segment, as opposed to a line, which minimizes the effect of leverage from residuals with large magnitudes,

far from the origin. In addition, the optimal line is the one that minimizes the Geman-McLure robust estimator of the perpendicular distance between the residuals and the line segment (as opposed to a more typical, but less robust Euclidean distance).

The line segment is parametrized using an angle θ, which determines $\mathbf{e} = [\cos\theta, \sin\theta]^\top$, as well as with two lengths $\beta \leq 0$ and $\gamma \geq 0$) that define the end points $\mathbf{p} = \beta\mathbf{e}$ and $\mathbf{q} = \gamma\mathbf{e}$, where the line segment finds residual support. Please note that while the frame super-index (^{f}) has been dropped from the notation, the estimate of an epipolar line segment is still done for each frame (except on the base frame, where the estimate is not necessary).

The largest contributor to the robustness of the estimate of the line segment parameters (θ, β and γ) is a voting parameter-estimation technique (similar to a Hough transform) that uses a radial histogram to accumulate votes from the residual data in support to a discrete set of potential Epipolar orientations and lengths. A schematic diagram of this histogram is shown in Figure 4.8. The size of the histogram bins grows exponentially in the radial direction, in an attempt to further limit potential leverage effects caused by residuals with large magnitudes. The furthest histogram bin is determined by the residual with the largest magnitude.

After populating the histogram, a gap finding mechanism (see Section 4.2.4) is used to clear the votes cast by residuals after a salient magnitude gap for each orientation of the histogram. A gap is salient if its size is considerably larger than the median gap between residuals. After clearing votes beyond any gaps, the orientation with largest support determines θ, and the furthest support points at either side of the origin determine β and γ.

At this stage, constrains for both degenerate motions (based on 2D Affine model) as well as for non-degenerate motions (based on the 2DAPE decomposition) are available. The next stage consists of using these constraints to identify which of the trajectories in $\hat{\mathbf{W}}$ do in fact move coherently (at least with respect the control points) with either model of rigid motion. We call this stage Inlier Detection, and is explained next.

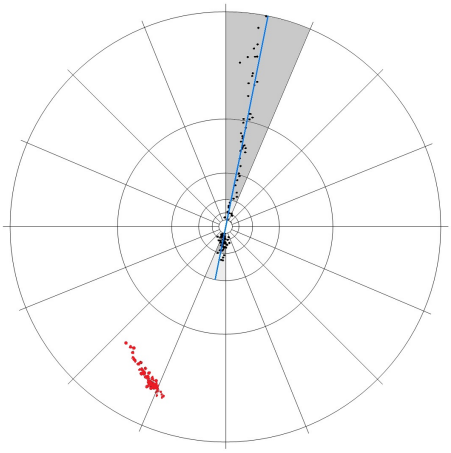


Figure 4.8: Voting histogram. Darker cell gray-level indicates larger support (vote count) for a particular epipolar direction and magnitude ranges. Note how the votes cast by the red trajectories are purposely not reflected with a darker cell, given that these votes lie behind a gap and were cleared before tallying.

4.2.4 Inlier Detection (B)

Inlier detection is possibly the most important step of each RANSAC trial, as it directly assesses the ability of the estimated motion model to explain the motion of other trajectories within $\hat{\mathbf{W}}$, and hence the overall quality of the motion model. This stage also has one of the largest effects in the evaluation of the objective function $\mathcal{O}(\hat{\mathbf{W}}, \mathbf{M}_i)$ from Equation 4.1 that determines the winning model from all RANSAC trials.

Formally, inlier detection is the process of determining the subset of trajectories whose motion can be predicted well by a model of motion. The result is a matrix $\mathbf{B} \in \{0,1\}^{F \times I}$ that represents whether the i^{th} trajectory is a model inlier at frame f ($b_i^f = 1$) or not ($b_i^f = 0$). Inlier detection is done differently depending on whether the presumed model of motion is degenerate or non-degenerate. Both mechanisms are explained next.

Degenerate Motion Inlier Detection

Suppose the matrix $\hat{\mathbf{W}}$ contains trajectories $\hat{\mathbf{W}}_1 \in \mathbb{R}^{2F \times I}$ and $\hat{\mathbf{W}}_2 \in \mathbb{R}^{2F \times J}$ from two independently moving, planar objects, and let these trajectories be contaminated by the Gaussian noise matrix $\mathbf{N} \in \mathbb{R}^{[2F \times I+J]}$ with zero-mean

and spherical covariance ($\mathbf{N} \sim \mathcal{N}(0, \sigma^2)$)

$$\hat{\mathbf{W}} = \left[\hat{\mathbf{W}}_1 | \hat{\mathbf{W}}_2\right] + \mathbf{N}. \tag{4.25}$$

Assume, also, that the true affine transformations $\mathbf{A}_1^f$ and $\mathbf{A}_2^f$ that describe the motions of $\hat{\mathbf{W}}_1$ and $\hat{\mathbf{W}}_2$ respectively, are known, as well as the exact location of the trajectories at an arbitrary base frame $\hat{\mathbf{w}}^b$. If $\mathbf{A}_1^f$ is used to compute predictions for all the trajectories in $\hat{\mathbf{W}}$ at frame f, the magnitude of the residual displacements for trajectories in $\hat{\mathbf{W}}_1$, namely $\hat{\mathbf{R}}_1 = [\mathbf{r}_1^f, \mathbf{r}_2^f, \ldots, \mathbf{r}_I^f]$, estimated using Equation 4.21, would be equal to the magnitude of the noise for that trajectory

$$|\mathbf{r}_i^f|^2 = |\mathbf{n}_i^f|^2, \tag{4.26}$$

and its expected value would be

$$E[|\mathbf{n}_i^f|^2] = 2\sigma^2. \tag{4.27}$$

The latter is true because $|\mathbf{n}_i^f|^2$ is a random variable that corresponds to the squared (Euclidean) norm of a two-dimensional vector of independent, normally-distributed random variables, and when its variance is normalized, it can be characterized with a Chi-squared distribution of $k = 2$ degrees of freedom:

$$\frac{|\mathbf{n}_i^f|^2}{\sigma^2} \sim \chi_2^2, \tag{4.28}$$

whose expected value is equal to the number of degrees of freedom:

$$E\left[\chi_2^2\right] = 2. \tag{4.29}$$

Removing the normalization (multiplying by σ^2) leads to the result of Equation 4.27.

On the other hand, trajectories from $\hat{\mathbf{W}}_2$ will be predicted using the wrong model ($\mathbf{A}_1^f$), resulting in residuals with magnitudes determined by the motion differential

$$\left|(\mathbf{A}_1^f - \mathbf{A}_2^f)\hat{\mathbf{w}}_i^b\right| = \left|\mathbf{r}_i^f\right|. \tag{4.30}$$

This analysis suggests that the two motions are distinguishable when the

motion differential is bigger than the average displacement due to noise

$$\left|(\mathbf{A}_1^f - \mathbf{A}_2^f)\mathbf{w}_i^b\right|^2 > 2\sigma^2. \tag{4.31}$$

Intuitively, Equation 4.31 is checking whether the magnitude of the displacement difference between the two independently moving objects is larger than the magnitude of the noise. In other words, when the above inequality holds, using $\mathbf{A}_1^f$ to make predictions of $\hat{\mathbf{W}}_1$ produces residuals $\hat{\mathbf{R}}_1$ whose expected magnitude is be equal to the magnitude of the noise: $2\sigma^2$, but using the same 2D Affine model ($\mathbf{A}_1^f$) to make predictions of the independently moving trajectories in $\hat{\mathbf{W}}_2$ produces residuals of a statistically larger magnitude.

Using this observation, model inliers can be determined by thresholding $|\mathbf{r}_i^f|$, where the threshold (τ) is the magnitude of the noise, scaled by a constant ($\tau = \lambda_\sigma \sigma$):

$$b_i^f = \begin{cases} 1, & |\mathbf{r}_i^f| \leq \tau \\ 0, & \text{otherwise.} \end{cases} \tag{4.32}$$

But because the value of σ is generally unknown, the threshold (τ) has to be estimated automatically from the residual data.

Figure 4.9 shows four frames of the residuals from a 2D Affine model ($\mathbf{A}^f$) that was estimated using three of the blue-class trajectories as the control points (using Equation 4.11). Color indicates ground truth class labeling. The magnitude of the noise of the inlier residuals (blue dots) is consistently smaller than the magnitude of the outlier residuals (green crosses). These differences in magnitude create a gap, and the threshold τ separates trajectories at either side of it. We found that τ could be estimated as the magnitude of the one residual that lies just before a salient magnitude gap, closest to the origin. A gap is salient if the magnitude difference between two consecutive residual magnitudes is at least λ_τ times the median distance between consecutive magnitudes.

Formally, to estimate τ, let $\hat{\mathbf{r}}$ be a vector of sorted residual magnitudes

$$\hat{\mathbf{r}} = [r_{p(1)}, r_{p(2)}, \ldots, r_{p(I)}] \tag{4.33}$$

where $p(i)$ is a permutation of $\{1, \ldots, I\}$ such that $|\hat{r}_{p(i)}| \leq |\hat{r}_{p(i+1)}|$, and let

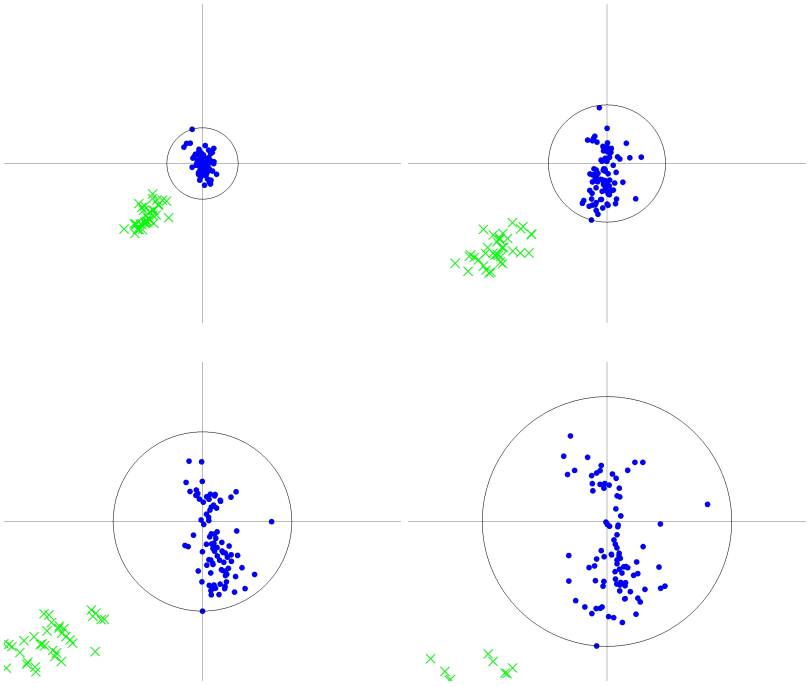


Figure 4.9: 2D model residuals at four different frames, only residuals that correspond to trajectories in $\hat{\mathbf{W}}$ are shown. Color indicates ground truth class labeling. Dots are model inliers, crosses are outliers. Gap threshold (τ) indicated as a black circle, centered at the origin. Notice the presence of a magnitude gap between the two clusters at each frame.

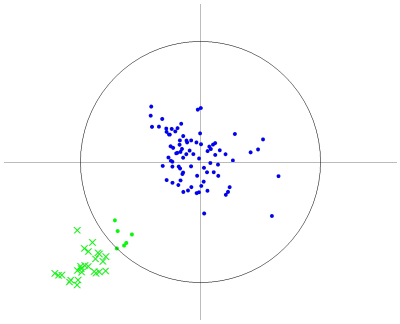
$\tilde{r} = \text{median}\left(\{|r_i|\}_{i=1}^{I}\right)$, then the threshold can be formally defined as

$$\tau = \min\{\hat{r}_i \mid (\hat{r}_{i+1} - \hat{r}_i) > \lambda_\tau \tilde{r}\}, \tag{4.34}$$

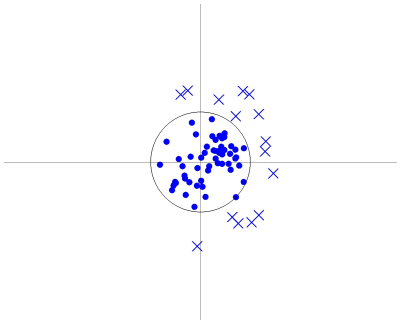
where the parameter λ_τ determines the smallest acceptable magnitude gap as a linear scaling of the residual median gap.

The black circles in Figure 4.9 show the estimate of τ at each frame using Equation 4.34. The inlier and outlier subsets are correctly identified on all four frames.

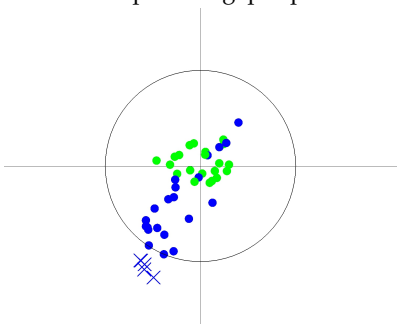
In contrast, Figure 4.10 shows three examples where the gap finding algorithm fails to accurately separate trajectories between the inlier and outlier classes. The first and possibly most common failure mode (Figure 4.10a) occurs when the inlier class trajectories (in blue) are contaminated with long-tail noise,



(a) Salient gap is contaminated by noisy trajectories from the blue class.



(b) Single class residuals. Spurious gap separates noisier residuals.



(c) Motion differences are indistinguishable from the underlying noise.

Figure 4.10: Failure modes for the 2D inlier detection algorithm. Color indicates ground truth class labeling. Dots are model inliers, crosses are outliers. Gap threshold (τ) indicated as a black circle, centered at the origin.

and at the same time, the outlier class trajectories (in green) have residuals with only marginally larger magnitudes. In this case, the long-tail noise of the inlier trajectories blurs the inlier-outlier gap, and the spurious one found the algorithm is already within the outlier class, rendering a contaminated inlier subset.

In the second example (Figure 4.10b) the inlier class (in blue) is also contaminated with long-tail noise, but in this case, the lower density of residuals at the skirts of the distribution creates a spurious gap, and while this situation renders an uncontaminated inlier subset (all the inliers are still from the right class), the output motion benefits from the largest possible number of trajectories.

The last scenario (Figure 4.10c) is due to a contaminated motion model, estimated from a set of mismatched control points where two came from the green set and one from the blue set. This contaminated model poorly predicts both motions, leading to very large residual magnitudes and, most importantly, indistinguishable inlier-outlier distributions, and while this misclassification cannot be attributed to the gap finding algorithm, it still renders a contaminated inlier subset.

Certainly, a more robust inlier detection mechanism could be used at this stage, but because the LMMF is run several times, and the gap finding algorithm is run for every RANSAC trial and for every frame, detection accuracy is compromised in the interest of computational efficiency, knowing that subsequent stages of the algorithm will aggregate independent sources information to discard contaminated inlier subsets.

Non-Degenerate Motion Inlier Detection

Unlike with the degenerate case, where the magnitude of the residuals is used, the non-degenerate inlier detection procedure is based on the 2DAPE decomposition, using a metric that evaluates proximity to the estimated Epipolar line segment.

Trajectory i becomes an inlier at frame f if it satisfies two conditions. First, the projection of $\mathbf{r}_i^f$ onto the estimated line segment must lie within the segment limits ($\beta \leq \mathbf{r}_i^{f\top} \mathbf{e}^f \leq \gamma$). Second, the normalized distance to the line must be below a threshold ($\mathbf{e}_\perp^{f\top} \mathbf{r}_i^f \leq \sigma_2 \lambda_d$), where $\mathbf{e}_\perp^f$ is the perpendicular direction to $\mathbf{e}^f$. Notice that the threshold depends on the smallest singular value from Equation 4.37 to (roughly) account for the presence of noise in the direction perpendicular to the epipolar line.

$\mathbf{e}_\perp^f$ Direction perpendicular to the Epipolar line at frame f.

The use of the 2DAPE decomposition as a constraint to evaluate whether trajectories align to a coherent rigid non-degenerate motion model relies on the assumption that a set of 2D Affine residuals that exhibit a tight linear

distribution is unlikely to be accidental, but in fact are the likely result of unaccounted epipolar displacements from a non-planar object undergoing a non-degenerate motion. It must be noted that the 2DAPE constraint is a much looser one compared to the one imposed by a full 3D Affine model. When using the 2DAPE model, the relative depth of each trajectory remains unconstrained (*i.e.*, it can vary at each frame), whereas a 3D Affine demands a fixed relative depth across all frames (albeit it may allow some unlikely global deformations or reflections). Still, our experimental results suggest that the 2DAPE-based inlier detection technique proposed here is suitable to evaluate rigid motion coherence.

Inlier Detection of Full Trajectories

The inlier detection procedures described in Section 4.2.4 above, determine whether a trajectory is an model inlier at frame f. A full trajectory (including all F frames) is an inlier if it is so for more than a fraction ($0 < \lambda_b < 1$) of the frames, as is indicated by the binary vector $\hat{\mathbf{b}}$

$\hat{\mathbf{b}}$ Binary vector of inl trajectories.

$$\hat{\mathbf{b}} = [\hat{b}_1, \hat{b}_2, \dots \hat{b}_I]^\top, \tag{4.35}$$

where

$$\hat{b}_i = \begin{cases} 1, & \left(\sum_{f=1}^{F} b_i^f\right) \geq \lambda_b F \\ 0, & \text{otherwise.} \end{cases} \tag{4.36}$$

In our implementation we set $\lambda_b = 0.5$.

Once the subset of inliers for either model is known, a rough estimate of the magnitude of the noise can be computed.

4.2.5 Estimating the Magnitude of the Noise

Keeping in mind that the goal of estimating the parameters of a model $\mathbf{M}_i$, given the control trajectory indices $\mathbb{D}_i$, is to evaluate $\mathcal{O}(\hat{\mathbf{W}}, \mathbf{M}_i)$, which quantifies the accuracy and efficiency of model $\mathbf{M}_i$ to make motion predictions for a subset of (inlier) trajectories. The model that maximizes the objective function from a set of K proposals becomes the output of the LMMF algorithm.

Estimating the magnitude of the noise for each frame (σ^f) enables the

σ^f Estimate of the mag tude of the noise at fra

use of a likelihood model in the objective function $\mathcal{O}(\cdot)$, as opposed to an Euclidean distance metric that would not account for the underlying level of noise.

Degenerate Case

To compute the magnitude of the noise σ^f of a degenerate model of motion, let $\hat{\mathbf{R}}^f$ be a $2 \times n$ matrix formed by stacking the residuals of the subset of 2D Affine inlier trajectories (those for which $\hat{b}_i = 1$, rendering $n = \sum_i b_i^f$) for the degenerate model at frame f, and let $\mathbf{USV}^\top$ be the singular value decomposition of the covariance matrix of $\hat{\mathbf{R}}^f$:

$$\mathbf{USV}^\top = \texttt{svd}\left(\frac{1}{\sum b_p^f}(\hat{\mathbf{R}}^f)^\top \hat{\mathbf{R}}^f\right). \tag{4.37}$$

The magnitude of the noise is defined as the largest singular value

$$(\sigma^f)^2 = s_1. \tag{4.38}$$

The motivation here is that, if the motion is in fact degenerate, then the only unaccounted-for displacements captured by the residuals are due to noise, which is assumed isotropic. In fact, when the inlier residual distribution is far from isotropic, the 2D model is most likely insufficient and an upgrade to 3D may be necessary, as explained in Section 4.2.2.

Non-Degenerate Case

Similarly to the 2D case, let $\hat{\mathbf{R}}^f$ contain the 2D Affine residual data, but now using the inlier labeling from the non-degenerate model, and let $\mathbf{S} = [s_1, s_2]$ (with $s_1 > s_2$) be the singular values of the corresponding covariance matrix, as in Equation 4.37.

In this case, the largest singular value s_1 captures the spread of residuals along the Epipolar line, so its magnitude is mainly related to the magnitude of the Epipolar displacements, due to relative depth. However, s_2 captures deviations in the perpendicular direction, which in a rigid 3D object can only be attributed to noise, making

$$(\sigma^f)^2 = s_2 \tag{4.39}$$

a reasonable estimate for its magnitude, assuming isotropic noise.

4.3 Objective Function

The objective $\mathcal{O}(\hat{\mathbf{W}}, \mathbf{M})$ quantifies the accuracy and efficiency with which model $\mathbf{M}$ makes motion predictions for a subset of inlier trajectories (indicated by $\hat{\mathbf{b}}$), noting that there is a deterministic mapping between the control trajectories indexed by $\mathbb{D}_i$ and the parameters of the model $\mathbf{M}_i$.

Having explained the parameters of $\mathbf{M}$, the objective is defined as:

$$\mathcal{O}(\hat{\mathbf{W}}, \mathbf{M}) = \sum_{f \in F} \sum_{i \in I} \hat{b}_i L_\omega \left(\hat{\mathbf{w}}_i^f \mid \mathbf{M} \right) + \lambda_\Psi \Psi(\omega) + \Gamma(\mathbf{B}), \tag{4.40}$$

where $\hat{\mathbf{w}}_i^f = (x_i^f, y_i^f)$ are the x and y coordinates of the i^{th} trajectory from the support subset $\hat{\mathbf{W}}$ at frame f. The function $L_\omega(\cdot)$ promotes prediction accuracy, and both $\Psi(\cdot)$ and $\Gamma(\cdot)$ are regularization terms that penalize model complexity and modelling insufficiency, respectively. All three terms are defined next.

The definition of the negative log-likelihood function $L_\omega(\cdot)$ depends on the estimated model type ($\omega = 2D$ or $\omega = 3D$).

For the degenerate case, $L_{2\text{D}}(\cdot)$ is defined as:

$$L_{2\text{D}}(\hat{\mathbf{w}}_i^f \mid \mathbf{M}) = -\log \left(\frac{1}{2\pi\sigma^f} \exp \left\{ -\frac{\mathbf{r}_i^{f\top} \mathbf{r}_i^f}{2(\sigma^f)^2} \right\} \right), \tag{4.41}$$

which is the negative logarithm of a zero-mean 2D Gaussian distribution evaluated at the residuals $\mathbf{r}_i^f$. The spherical covariance matrix is defined as $\Sigma = (\sigma^f)^2 \mathbf{I}$. The values for the residuals $\mathbf{r}_i^f$ and for the estimates of the magnitude of the noise σ^f are computed using Equations 4.21 and 4.37, respectively.

On the other hand, the accuracy function for the non-degenerate case $L_{3\text{D}}(\cdot)$ is based on the 2DAPE decomposition and is defined as follows:

$$L_{3\text{D}}(\hat{\mathbf{w}}_i^f \mid \mathbf{M}) = -2\log \left(\frac{1}{\sqrt{2\pi}\sigma^f} \exp \left\{ -\frac{\left(\mathbf{r}_i^{f\top} \mathbf{e}_\perp^f \right)^2}{2(\sigma^f)^2} \right\} \right), \tag{4.42}$$

which is also the negative logarithm of a zero-mean 2D Normal distribution

computed as the product of two identical, separable, single-variate, normal distributions, evaluated at the distance from the residual to the Epipolar line (note the factor of 2 in front of the right hand side of equation 4.42). One of these two contributions corresponds to the actual deviation in the direction of $\mathbf{e}_{\perp}^{f}$, which is analytically computed using $\mathbf{r}_i^{f\top}\mathbf{e}_{\perp}^{f}$. The second one corresponds to an estimate of the deviation in the perpendicular direction ($\mathbf{e}^{f}$), which cannot be determined using the 2DAPE decomposition model, but can be approximated to be equal to $\mathbf{r}_i^{f\top}\mathbf{e}_{\perp}^{f}$, which is a reasonable estimate under the isotropic noise assumption. The likelihoods of this objective function are thus comparable to those produced by $L_{2\mathrm{D}}(\cdot)$.

The function $\Psi(\omega)$ penalizes model complexity. With this in mind, the 2D Affine model used to compute model predictions for the degenerate case requires estimating 6 parameters per frame, except for the base frame. In contrast, the 3D Affine model that would be necessary for the non-degenerate case requires 8 parameters, plus an extra parameter to account for the relative depth of each trajectory in each frame (other than the base frame). We select the model complexity cost to be proportional to the number of free parameters in the motion model. Therefore for a trajectory dataset $\hat{\mathbf{W}} \in \mathbb{R}^{[2F\times I]}$, the model complexity penalty term is:

$$\Psi(\omega) = \begin{cases} 6(F-1), & \text{if } \omega = 2\mathrm{D} \\ 8(F-1)+I, & \text{if } \omega = 3\mathrm{D}. \end{cases} \tag{4.43}$$

Finally, the function $\Gamma(\mathbf{B})$ (strongly) penalizes models that describe too few trajectories using:

$$\Gamma(\mathbf{B}) = \begin{cases} \infty, & \text{if } \sum_i \hat{b}_i < \lambda_\Gamma \\ 0, & \text{otherwise} \end{cases} \tag{4.44}$$

with $\hat{\mathbf{b}}$ determined from $\mathbf{B}$ using Equation 4.36. In our implementation we use $\lambda_\Gamma = 5$.

4.4 Other Considerations of the LMMF Algorithm

This section includes details about the LMMF algorithm that despite being relevant to specific sections of this chapter (in particular to the 2D Affine Model Instantiation, Inlier Detection and Noise Estimation sections), have

only been included at the end of the current chapter to improve presentation coherence and readability.

Contaminated 2D Affine Models

Because of the random nature of the sampling procedure, noisy trajectories, tracking failures and trajectories from independently moving objects will eventually become part of a set of control trajectories. This section describes four normal scenarios where the (randomly drawn) subset of control trajectories renders a contaminated 2D Affine model. The goal of presenting these cases is to give some intuition about the failure modes of the LMMF algorithm.

Keep in mind that these candidate motion models live in the context of a random-sampling, model-instantiation procedure, where only the model that maximizes the objective function $\mathcal{O}(\cdot)$, (Equation 4.1), will survive to become an element of the set of candidate models of motion $\mathbb{C}$.

For clarity of explanation, please assume that there is in fact a subset of trajectories within $\hat{\mathbf{W}}$ whose motion can be explained with a 2D Affine model.

The first scenario has to do with the presence of noise in the control points. This noise corrupts the estimate of the parameters of $\mathbf{A}^f$ (Equation 4.11) in a way that is proportional to its magnitude, leading to poor prediction accuracy, and consequently, to sub-optimal evaluations by the objective function, especially when compared to other less noisy models. This naturally prevents noisy models from becoming the output of the LMMF algorithm.

The second scenario considers the presence of tracking failures (drifts, trajectories going out of frame, trajectories undergoing self occlusion, etc.) as part of the control points, in which case the resulting model fails to make good predictions for any of the motions in the scene. This situation leads to very large prediction errors, allowing the optimization mechanism to discard these models naturally as well.

The third scenario occurs when each trajectory lies on an independently moving object, with almost identical consequences to the previous case, as the resulting model is completely incapable of making good predictions for many of the trajectories in $\hat{\mathbf{W}}$.

Finally, the fourth case occurs when all three control-point trajectories are successfully tracked throughout the sequence, but only two of them lie on the same independently moving object. This can be a difficult case to

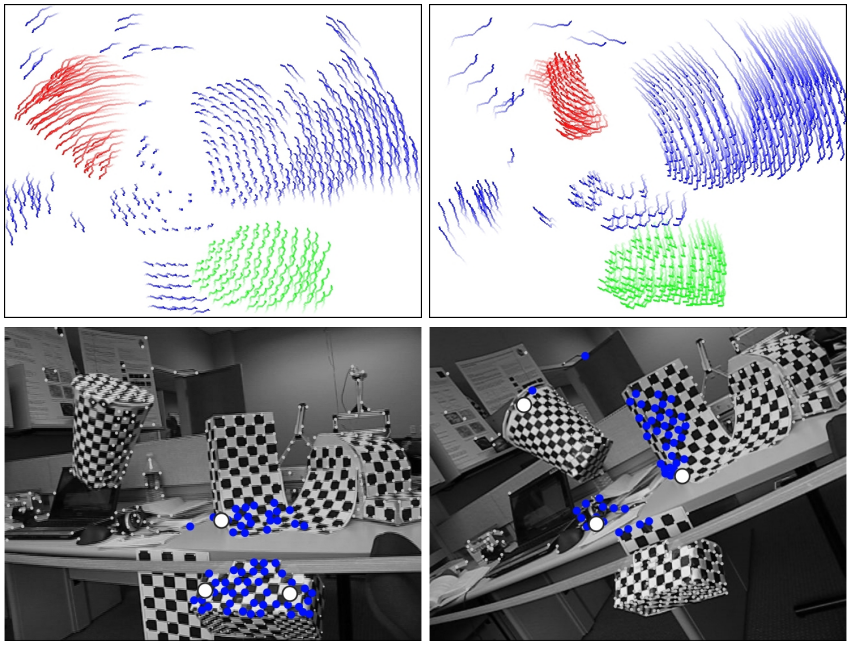


Figure 4.11: Two failure cases. Control trajectories originate from mismatched independent objects, but the resulting predictions are relatively accurate. The top row shows full trajectories color coded with ground truth. The bottom row shows the mismatched set of control trajectories (white dots), and the resulting inlier subsets (blue dots) for each case containing trajectories from more than one class.

detect, because depending on the spatial disposition of the trajectories and the characteristics of each motion, the estimated 2D Affine transformations will occasionally make good predictions for a small subset of trajectories within a small image region. This constitutes one of the main failure modes of the LMMF algorithm. Figure 4.11 shows two examples, where mismatched control trajectories render inlier subsets with trajectories that belong to different classes.

Inlier Count vs Inlier Accuracy

In the original RANSAC method [12], candidate models are acceptable if they describe at least as many data points as indicated by a fixed threshold, and within these, the model with the maximum inlier count is preferred. In contrast, in our RANSAC objective (Equation 4.40), the size of the inlier

subset is not as relevant as the accuracy with which inlier trajectories are modeled: note how the inlier binary variables multiply the likelihood term in Equation 4.40, which implies that the accuracy-related penalty paid by a noisy trajectory can be saved by simply labelling the trajectory as an outlier, and while this suggests that the optimal model is one with no inliers, two mechanisms prevent this from happening. The first one is by including the $\Gamma(\mathbf{B})$ term, defined in Equation 4.44, which discards models that describe fewer trajectories than a fixed threshold (although in our implementation this threshold is only marginally larger than the minimum number of trajectories needed to estimate the motion), much like the standard RANSAC method. The second reason is simply that the objective function $\mathcal{O}(\hat{\mathbf{W}}, \mathbf{M})$ in Equation 4.40 is never optimized over the inlier variables ($\mathbf{B}$). Instead, all the parameters of the model (including $\mathbf{B}$) are determined by a deterministic mapping $(\hat{\mathbf{W}}, \mathbb{D}_i) \rightarrow \mathbf{M}_i$ from the local set of trajectories ($\hat{\mathbf{W}}$), as well as the three selected control points ($\mathbb{D}_i$). These inlier trajectories then directly specify the motion and noise parameters for the model ($\mathbf{M}_i$).

On the objective function

It is worthwhile to briefly take a step back and consider the purpose of this objective function. The key issue at this stage is to select the most promising individual motion proposals from within a large list. Several of the most promising proposals will then be combined (see the next chapter) to form a model for multiple moving objects in the scene. Therefore, the key property of any individual model is its ability to extrapolate to fit all the trajectories in a single rigidly moving component. Of course, the correct segmentation result is unavailable at this stage, so evaluation of this property directly is not possible.

We tried to estimate extrapolation accuracy by first estimating the inliers for a single motion model, along with the noise magnitude, and then using these estimated inliers to measure the accuracy. However, we found it difficult to get a sufficiently accurate joint estimate of both the inlier set and the noise magnitude, especially given the presence of other rigid motions in the data. Therefore we instead chose to use the current conservative approach for inlier assignment, and postpone the accurate classification of all trajectories until we have models for all (or at least the majority) of the motions present.

This still left us with the problem of predicting the extrapolation accuracy of a single model, knowing only a subset of the inliers. We observed empirically that sparser inlier subsets (with smaller estimated noise variances) could render models that make better extrapolation predictions. The above objective function reflects this observation. In particular, tight models with small variances, populated by smaller subsets of inliers, can result in better (i.e., smaller) objective function values, compared to models with a large inlier subset but with relatively larger variances. It is possible that this observation is a consequence of the type of noise in our experimental datasets, and this issue could benefit from further investigation.

Noise

Random sampling of model control points from a small, spatially-local region results in a trade-off between the likelihood of them belonging to the same class versus extrapolation error.

A motion model is extrapolated when used to make predictions of the motion of trajectories outside the region bounded by the control points, and the magnitude of modeling error often depends on the magnitude of the noise in the control points. In particular, extrapolation error, although hard to characterize in general, can be expected to grow with the distance from the location of the prediction to the nearest control point, and inversely with the distance between the control points themselves. Figure 4.12 illustrates this phenomenon by comparing the distributions of noisy predictions that originate from four different scenarios with varying levels of extrapolation distances. A total of 500 predictions are made for each of the 16 target points using a 2D Homography estimated from three control points. For each prediction, one of the control points is contaminated with zero-mean Gaussian noise of small magnitude. Identical noise is added to the same trajectory in all four scenarios. Resulting distributions illustrate how the variance of increases as the distance between the control points decreases.

The affine projection model computed in Section 4.2.1 certainly suffers from the extrapolation effects of noise, and does so particularly because its parameters are computed using trajectories from a locally coherent neighborhood. During LMMF, this is not a real problem because the predicted trajectories also lie within the locally coherent neighborhood of the control

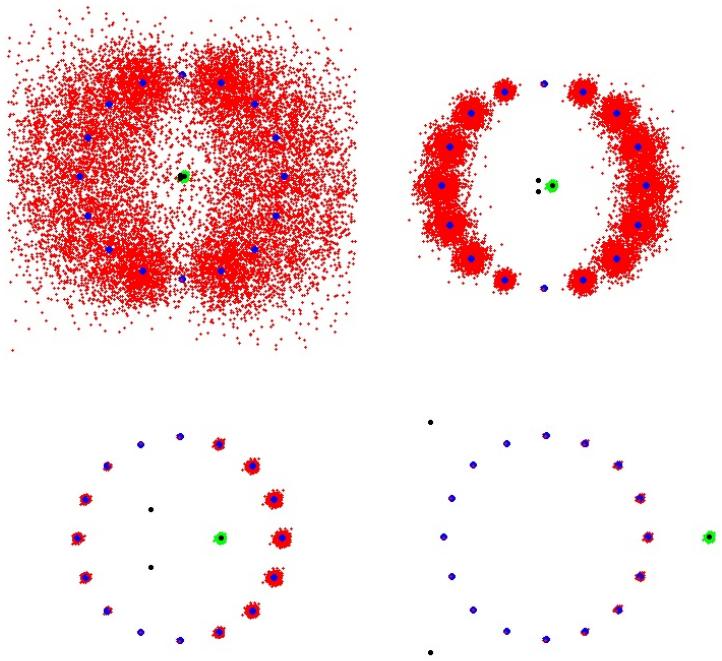


Figure 4.12: Predictions (red) from a 2D Affine model computed from 3 control points (black) where one was contaminated with standard Gaussian noise (green). Noiseless model predictions in blue. All four scenarios have identical noise. Clearly, the magnitude of the extrapolation error changes with the distance between the control points.

points. However, in future stages of the motion segmentation algorithm, the estimated model of motion will be used to make motion predictions for trajectories of the entire dataset, potentially far from the control points, where accuracy may prove insufficient and must therefore be upgraded to include additional information that decreases the effects of white noise. The orthonormal basis estimated in Section 5.3.1 reduces the effects of noise in the control points by using a least squares fit to all of the trajectories deemed to be inliers.

4.5 Algorithm Pseudo-Code

A pseudo-code description of the LMMF algorithm can be found in Algorithm 4.1.

Algorithm 4.1: Local Motion Model Fitting

Input: Locally-coherent trajectory data $\hat{\mathbf{W}}_{[2F\times I]}$, number of RANSAC trials K, arbitrary base frame b with $1 \leq b \leq F$.

Output: Parameters of the motion model $\mathbf{M} = (\mathbb{A}, \mathbb{E}, \mathbf{B}, \boldsymbol{\sigma}, \omega)$ and matrix of inlier trajectories $\mathbf{W}_{\hat{\mathbf{b}}}$

$l^\star = \infty$ *// initialize best neg-loglikelihood*

$\mathbf{X} \leftarrow \texttt{homogeneousCoords}(\hat{\mathbf{W}}^b)$ *// points at base frame*

for $k \in \{1, \ldots, K\}$ **do**

  $\mathbb{D} \leftarrow \texttt{rand}(3, [1, I])$ *// three random control trajectory indices*

  for $f \in \{1, \ldots, F\} - \{b\}$ **do**

    $\mathbf{Y} \leftarrow \texttt{homogeneousCoords}(\hat{\mathbf{W}}^f)$ *// points at frame f*

    $\mathbf{X}_{\mathbb{D}} \leftarrow \texttt{selectColumns}(\mathbf{X}, \mathbb{D})$ *// control trajectory data*

    $\mathbf{Y}_{\mathbb{D}} \leftarrow \texttt{selectColumns}(\mathbf{Y}, \mathbb{D})$

    $\mathbf{A}^f \leftarrow \mathbf{Y}_{\mathbb{D}}\mathbf{X}_{\mathbb{D}}^{-1}$ *// 2D Affine model*

    $\mathbf{R} \leftarrow \mathbf{A}^f\mathbf{X} - \mathbf{Y}$ *// residuals*

    $[\mathbf{B}^f_{2\mathrm{D}}, \sigma^f_{2\mathrm{D}}] \leftarrow \texttt{compute2DInliers}(\mathbf{R})$ *// inliers*

    $\mathbf{L}^f_{2\mathrm{D}} \leftarrow \texttt{compute2DNegLogLikelihoods}(\mathbf{R}, \sigma^f_{2\mathrm{D}})$ *// 2D likelihoods*

    $[\mathbf{U}, \mathbf{S}, \mathbf{V}^\top] = \texttt{svd}(\texttt{weightedCov}(\mathbf{R}, \mathbf{B}_{2\mathrm{D}})$ *// s_1 and s_2*

    if $\frac{s_1}{s_2} > 1 + \lambda_\omega$ **then**

      $[\mathbf{B}^f_{3\mathrm{D}}, \mathbf{e}^f, \sigma^f_{3\mathrm{D}}] \leftarrow \texttt{compute3DInliers}(\mathbf{R})$

      $\mathbf{L}^f_{3\mathrm{D}} \leftarrow \texttt{compute3DNegLogLikelihoods}(\mathbf{R}, \sigma^f_{3\mathrm{D}})$

  // determine trajectory inliers

  for $i \in \{1, \ldots, I\}$ **do**

    $\hat{\mathbf{b}}_{2\mathrm{D}}(i) = \left(\sum_f \mathbf{B}_{2\mathrm{D}}(f, i) > \lambda_b F\right)$

    $\hat{\mathbf{b}}_{3\mathrm{D}}(i) = \left(\sum_f \mathbf{B}_{3\mathrm{D}}(f, i) > \lambda_b F\right)$

  // complete penalized neg-loglikelihoods

  $l_{2\mathrm{D}} \leftarrow \sum_f \sum_i \hat{\mathbf{b}}_{2\mathrm{D}}(i)\mathbf{L}_{2\mathrm{D}}(f, i) + \lambda_\Psi \Psi(2\mathrm{D}) + \Gamma(\mathbf{B}_{2\mathrm{D}})$

  $l_{3\mathrm{D}} \leftarrow \sum_f \sum_i \hat{\mathbf{b}}_{3\mathrm{D}}(i)\mathbf{L}_{3\mathrm{D}}(f, i) + \lambda_\Psi \Psi(3\mathrm{D}) + \Gamma(\mathbf{B}_{3\mathrm{D}})$

  // keep the best model overall

  if $(\min(l_{2D}, l_{3D}) < l^\star)$ **then**

    $\mathbf{M}^\star \leftarrow \mathbf{M}$

    if $(l_{2D} < l_{3D})$ **then**

      $l^\star \leftarrow l_{2\mathrm{D}}$

      $\mathbf{W}_{\hat{\mathbf{b}}} = \texttt{selectColumns}(\hat{\mathbf{W}}, \hat{\mathbf{b}}_{2\mathrm{D}})$

    else

      $l^\star \leftarrow l_{3\mathrm{D}}$

      $\mathbf{W}_{\hat{\mathbf{b}}} = \texttt{selectColumns}(\hat{\mathbf{W}}, \hat{\mathbf{b}}_{3\mathrm{D}})$

return $\mathbf{M}^\star$

5 Multiple Motion Model Fitting

This chapter describes the Multiple Motion Model Fitting (MMMF) framework to solve the problem of modeling trajectory data from orthographic scenes with multiple independently-moving rigid objects.

The method is based on the assumption that the motion of trajectories that arise from N independently moving objects can be modeled with a N-tuple of independent models of rigid motion

$$\mathbb{T} = (\mathbf{M}_1,\ \mathbf{M}_2, \ldots, \mathbf{M}_N), \tag{5.1}$$

plus a matrix $\mathbf{L} \in \{0,1\}^{[I \times (N+1)]}$ that uses a one hot encoding to indicate trajectory labels (as either of the N inlier class or as the outlier class). It is also assumed that the number of independent motions N is known (although an algorithm to estimate this number is proposed and evaluated in Section 7.3).

It is further assumed that each of the models in $\mathbb{T}$ can be instantiated locally, using the LMMF algorithm of Chapter 4, and that these local models are capable of extrapolating the motion of the majority of the trajectories (not necessarily within a spatially-local region) of the same independently moving object.

However, because the labeling of trajectories is obviously unknown a priori, and because the output of the LMMF algorithm is not always correct, it is unlikely that N models will in fact result in an appropriate set $\mathbb{T}$ in exactly N runs. Instead, the MMMF algorithm first builds a set of M candidate models

M Number of candidate motion models in $\mathbb{C}$.

$$\mathbb{C} = \{\mathbf{M}_1,\ \mathbf{M}_2, \ldots, \mathbf{M}_M\} \tag{5.2}$$

with many more candidates than motions in the scene ($M \gg N$), and then finds the subset $\mathbb{T} \subset \mathbb{C}$ that describes the whole trajectory data ($\mathbf{W}$) with maximum prediction accuracy, regularized by model complexity and modeling overlap.

The above argument suggests that the MMMF algorithm must solve two problems. One of coverage, where the selection of the spatially local subsets of trajectories ($\hat{\mathbf{W}}_i$) used to instantiate each candidate motion $\mathbf{M}_i \in \mathbb{C}$ must maximize sampling coverage to increase the chances of modelling every object in the scene (at least once, although having several is often better). And another one of model selection, where it must rank or score different model combinations $\mathbb{T} \subset \mathbb{C}$ with $|\mathbb{T}| = N$, in order to find the most suitable one.

The coverage problem is referred to as *Locally-Coherent Region Sampling* and the proposed solution is described in Section 5.1. The solution to the *Model-Combination Selection* problem is presented in Section 5.2.

5.1 Locally-Coherent Region Sampling

The goal of this section is to generate a rich set $\mathbb{C}$ of motion model candidates, with the intention of instantiating at least one but probably many motion models for each independently moving object. The method proceeds iteratively. During the i^{th} iteration, the subset of trajectories that lie within a spatially-coherent region becomes the support set $\hat{\mathbf{W}}^{(i)}$ (a sub-matrix with only a subset of the columns of $\mathbf{W}$, see next section). This trajectory data is then used as the input to the LMMF algorithm. The output of LMMF gives the parameters of the estimated motion model $\mathbf{M}_i$ and the subset of inliers $\hat{\mathbf{W}}^{(i)}_{\hat{\mathbf{b}}} \subseteq \hat{\mathbf{W}}^{(i)}$ (the trajectories actually explained by $\mathbf{M}_i$). The set $\hat{\mathbf{W}}^{(i)}_{\mathbf{b}}$ is referred to as the coverage of model $\mathbf{M}_i$.

Spatially local subsets of trajectories are defined by disk-shaped regions because they are simple to parametrize and maximally compact. With this type of region, the problem of sampling reduces to determining two parameters: a center and a radius.

5.1.1 Estimating the Locally-Coherent Region Parameters

Because there is no obvious benefit to modelling an arbitrary image region more frequently than any other, our sampling technique is designed to distribute modeling coverage as uniformly as possible, ideally resulting in each

trajectory being explained by (approximately) the same number of motion models.

With this in mind, the center of a region is chosen stochastically, using a non-parametric probability distribution $p(\mathbf{w}_k)$ over trajectories. This distribution is a function of how often $\mathbf{w}_k$ has been explained by all the models instantiated so far, and determines the likelihood of $\mathbf{w}_k$ to become the center of a new candidate region. So, if we define β_k as the number of times trajectory k has been a model inlier, then the distribution $p(\mathbf{w}_k)$ can be computed using

β_k Number of times trajectory k has been a model inlier.

$$p(\mathbf{w}_k) = \frac{(\lambda_d)^{\beta_k}}{\sum_j (\lambda_d)^{\beta_j}}, \tag{5.3}$$

where $0 < \lambda_d \leq 1$ is a constant that indicates how much relative likelihood a trajectory loses every time it is explained. Note that when $\beta_k = 0$ for all k, $p(\mathbf{w}_k)$ corresponds to a uniform distribution. In our implementation we set $\lambda_d = 0.1$.

Assuming $\mathbf{w}_k$ is randomly selected using the distribution of Equation 5.3, its image coordinates (x_k^f, y_k^f) at a uniformly sampled frame $f \sim \mathcal{U}(1, F)$ determine the center point

$$(o_x, o_y) = (x_k^f, y_k^f) \tag{5.4}$$

of the disk region. Note that while $\mathbf{w}_k$ is guaranteed to be in the resulting $\hat{\mathbf{W}}$, it is not guaranteed to be an inlier of the resulting motion model. That is to be determined by the LMMF alone.

Now, without any prior knowledge about the scale of the objects in the scene, determining a fixed size disk radius r is unlikely to work in general. Instead, the issue is avoided by randomly sampling disk-shaped regions of varying sizes from a uniform distribution $r \sim \mathcal{U}(\lambda_r, \lambda_R)$. In our implementation the value of λ_r is 10 pixels, and the value of λ_R is 0.15 times the width of the whole image.

Now, with the center (o_x, o_y), the radius r, and the base frame f, the support matrix of locally coherent data is defined as

$$\hat{\mathbf{W}} = [\mathbf{w}_{j_1}\ \mathbf{w}_{j_2}\ \ldots\ \mathbf{w}_{j_I}] \tag{5.5}$$

where $\{j \in \{1, \ldots, P\} \mid (x_j^f - o_x)^2 + (y_j^f - o_y)^2 < r^2\}$, and (x_j^f, y_j^f) are the x

and y coordinates of the point on trajectory $\mathbf{w}_j$ at frame f. Note that selected trajectories need to be within the disk region only at frame f. Also note that the construction of $\hat{\mathbf{W}}$ does not incorporate any knowledge about the motion of objects in the scene, and in consequence $\hat{\mathbf{W}}$ will likely contain trajectories that originate from more than one independently moving object (as shown in Figure 4.1). Finally, note that in a region that contains I trajectories, the input data for the LMMF algorithm is

$$\hat{\mathbf{W}} \in \mathbb{R}^{2F \times I}.$$

At each iteration, the frequencies β_k of Equation 5.3 are updated using the resulting inlier trajectory labels $\hat{\mathbf{b}}$ and the process is repeated to instantiate the M motion model proposals needed to populate the set $\mathbb{C}$ of Equation 5.2 (see Algorithm 5.1 for a pseudo-code description).

A final consideration is necessary to deal with outlier trajectories (*i.e.*, trajectories that drift from the original target, that go out of frame, that become self occluded or that become anomalous for any reason). This special treatment is necessary because these type of trajectories rarely become inliers from any of the motion models. And while this is a desirable behavior, a side consequence is that their corresponding inlier counters m_k remain at zero, or close. Then, after several motions have been instantiated, the disparity between large and close-to-zero inlier counts gives the corrupted and correctly under-modeled trajectories a disproportionately large likelihood that results in a pathological bias in the sampling scheme. This issue is alleviated by resetting $\beta_k = 0$ for all k every time a fixed number (λ_m) of disk regions is sampled (see Algorithm 5.1). In our implementation we use $\lambda_m = 10$.

5.2 Model-Combination Selection

At this stage, we assume that a set of candidate motion models $\mathbb{C}$ is available and the goal now is to explain the trajectories of all N independently moving rigid objects in the scene using a subset $\mathbb{T} \subset \mathbb{C}$, of N models of motion. The problem is posed as a discrete optimization one where the resulting N-combination of motion models is the one that optimizes an objective function that promotes prediction accuracy and penalizes model complexity and modeling overlap.

Algorithm 5.1: Disk Sampling

Input: Trajectory data $\mathbf{W}$, number of candidate models M, likelihood reset parameter $\lambda_m = 10$, likelihood decay parameter $\lambda_d = 0.1$
Output: A set of M candidate motion models $\mathbb{C} = \{\mathbf{M}_1, \mathbf{M}_2, \cdots \mathbf{M}_M\}$

$\boldsymbol{\beta} = \mathbf{0}$ *// Initialize coverage counters*
$\mathbb{C} = \emptyset$ *// Initialize the set of candidate models*
for $m = 1$ *to* M **do**
  $i =$`RandomSampling` $(\boldsymbol{\beta}, \lambda_d)$ *// Randomly sample for an index*
  $(o_x, o_y) = (x_i^f, y_i^f)$ *// Determine the origin*
  `r=rand` (r_{min}, r_{max}) *// Randomly sample for a radius*
  $\hat{\mathbf{W}}$=`TrajectoriesWithin` $(\mathbf{W}, o_x, o_y, r)$ *// Determine the spatialy local set*
  $[\mathbf{M}, \hat{\mathbf{b}}]$=`LMMF` $(\hat{\mathbf{W}})$ *// Instantiate a motion model using LMMF*
  $\mathbb{C} = \mathbb{C} \cup \mathbf{M}$ *// Add the model to the set*
  if `mod` $(m, \lambda_m) \neq 0$ **then**
    $\boldsymbol{\beta} = \boldsymbol{\beta} + \hat{\mathbf{b}}$ *// Update the coverage counters*
  else
    $\boldsymbol{\beta} = \mathbf{0}$ *// Reset coverage counters*
return $\mathbb{C}$

For the purpose of explaining the solution to this problem, let

$$\mathbb{T}_j = \left(\mathbf{M}_{q(j,1)}, \mathbf{M}_{q(j,2)}, \ldots, \mathbf{M}_{q(j,N)}\right) \tag{5.6}$$

be the j^{th} motion model combination, where $q(j,k)$ for $k \in \{1, \ldots, N\}$ indicates the index for the k^{th} individual motion model in $\mathbb{C}$ for this j^{th} combination.

All possible combinations can then be arranged as:

$$\mathbb{S} = \left(\mathbb{T}_1, \mathbb{T}_2, \ldots \mathbb{T}_{\binom{M}{N}}\right), \tag{5.7}$$

noticing that the size of $\mathbb{S}$

$$|\mathbb{S}| = \binom{M}{N} = \frac{M!}{N!(M-N)!)} \tag{5.8}$$

is potentially very large, as it grows nearly exponentially with the number of independent motions in the scene N.

The combinatorial optimization problem can thus be written as

$$\mathbb{T}^{\star} = \underset{\mathbb{T}_j \in \mathbb{S}}{\operatorname{argmin}}\, \mathcal{O}_s(\mathbb{T}_j), \tag{5.9}$$

where the objective loss function is

$$\begin{aligned} \mathcal{O}_s(\mathbb{T}_j) = & \sum_{n=1}^{N} \sum_{p=1}^{P} l_{p,n} E\left(\mathbf{w}_p, \mathbf{M}_{j_n}\right) \\ & + \lambda_{\Phi} \sum_{i=1}^{P} \Phi(\mathbf{w}_p, \mathbb{T}_j) + \lambda_{\Psi} \sum_{n=1}^{N} \Psi(\mathbf{M}_{j_n}). \end{aligned} \tag{5.10}$$

The first term in Equation 5.10 is a function that quantifies prediction error. Binary ownership variables $\left(\mathbf{L} \in \{0,1\}^{[P \times N]}\right)$ indicate whether the p^{th} trajectory is explained by the n^{th} model ($l_{p,n} = 1$) or not ($l_{p,n} = 0$), and are determined by the model with minimum prediction error:

$$l_{p,n} = \begin{cases} 1, & \text{if } \mathbf{M}_{j_n} = \underset{\mathbf{M} \in \mathbb{T}_j}{\operatorname{argmin}}\, E(\mathbf{w}_p, \mathbf{M}) \\ 0, & \text{otherwise.} \end{cases} \tag{5.11}$$

For simplicity, such deterministic mapping between a motion combination $\mathbb{T}_j$ and its corresponding labeling $\mathbf{L}_j$ is referred to as the model-to-labels mapping:

$$\mathbf{L}_j = \mathcal{F}(\mathbb{T}_j). \tag{5.12}$$

The rest of this chapter describes the remaining terms in Equation 5.10 and justifies the design choices. The prediction-error scoring function $E(\mathbf{w}, \mathbf{M})$ is presented first (Section 5.3), followed by the model overlap and model complexity regularization terms (Section 5.4).

5.3 Prediction-Error Scoring Function

The Prediction-Error Scoring Function $E(\cdot)$ plays a critical role in the MMMF algorithm as it strongly influences the outcome of the objective function $\mathcal{O}_s(\cdot)$ and consequently, the selection of the optimal $\mathbb{T}^{\star}$ (Equation 5.9) from a potentially very large set of possible model combinations. Crucially, $E(\cdot)$ also determines the motion segmentation labeling of trajectories, ac-

cording to highest prediction accuracy from each model in the combination (Equation 5.11). It is therefore desirable that this function has the following properties.

1. Given a large set S of model combinations $\mathbb{T}_i$, the overall ordering produced by $\mathcal{O}_s(\cdot)$ should give a better score to model combinations that more accurately explain the motion of trajectories from all N independent objects, and with the least number of model parameters.

2. If ground truth motion models were available and one formed the ground-truth model combination $\mathbb{T}^{GT} = \{\mathbf{M}_1^{GT}, \mathbf{M}_2^{GT}, \ldots, \mathbf{M}_N^{GT}\}$, the resulting labeling from Equation 5.11 should be very close to the correct (ground truth) labeling.

3. The estimation of $E(\cdot)$ for each combination should be computationally efficient, especially if many (and potentially up to $\binom{M}{N}$) model combinations are evaluated.

With these desirable properties in mind, we experimented with different ways of defining $E(\mathbf{w}, \mathbf{M})$. We found that the largest benefit comes from an accurate characterization of the underlying trajectory noise, allowing to correctly evaluate the residual distribution of each model.

It has been previously noted that the 2D Affine residuals can potentially be contaminated due to noise in the control trajectories. To prevent this situation, an alternative way of modeling motion is used, where the entire subset of inlier trajectories for each motion ($\hat{\mathbf{W}}_{\hat{\mathfrak{b}}}$) is used to estimate the parameters of the motion model at each frame (as opposed to the mere three control points indicated by $\mathbb{D}$), as explained next.

5.3.1 Orthonormal-Basis Residuals

In order to estimate a motion model with better extrapolation accuracy, the factorization method [45] is used, where a matrix of trajectories $\mathbf{W}$ is separated into motion ($\mathbf{P}$) and structure ($\mathbf{Q}$) matrices $\mathbf{PQ} = \mathbf{W}$. The method is equivalent to a least-squares reconstruction of $\mathbf{W}$ from the product of two rank-limited factors $\mathbf{P}$ and $\mathbf{Q}$. Because of its least-squares nature, it naturally deals with zero-mean Gaussian noise and benefits from a large number of trajectories, but

for the same reason, the parameter estimates could be severely contaminated by the presence of outliers.

The binary vector $\hat{\mathbf{b}}$ readily indicates the subset of (disk bounded) inlier trajectories that should be factorized to estimate the motion orthonormal basis $\mathbf{P}$. So, let

$$\hat{\mathbf{W}}_{\hat{\mathbf{b}}} = \begin{bmatrix} \hat{\mathbf{W}}_{q(1)} & \hat{\mathbf{W}}_{q(2)} & \cdots & \hat{\mathbf{W}}_{q(n)} \end{bmatrix} \text{ with } \hat{\mathbf{b}}_{q(i)} = 1, \tag{5.13}$$

be the $2F \times n$ matrix of inlier trajectories, and let $\bar{\mathbf{w}}_{\hat{\mathbf{b}}}$ be the $2F$-vector that contains the mean of the inlier trajectories (those in $\hat{\mathbf{W}}_{\hat{\mathbf{b}}}$) for all frames.

Then, the orthonormal basis $\mathbf{P}$ of a $\omega = 2$D (or 3D) motion model can be determined by the 2 (or 3) left singular vectors of the mean-subtracted inlier trajectories, as in:

$\mathbf{P}$ Orthonormal basis motion model.

$$\mathbf{U}\Sigma\mathbf{V}^\top = \texttt{svd}(\mathbf{W}_{\hat{\mathbf{b}}} - \bar{\mathbf{w}}_{\hat{\mathbf{b}}}). \tag{5.14}$$

So, if $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{2F}]$, then the matrix $\mathbf{P}$ is defined as $\mathbf{P} = [\mathbf{u}_1, \mathbf{u}_2]$ for $\omega = 2D$ and as $\mathbf{P} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$ for $\omega = 3D$.

Residuals

The matrix $\mathbf{P}$ can then be used to make a prediction $\mathbf{p}$ for a mean-subtracted trajectory $\mathbf{w}$ by projecting it onto the rank-limited subspace and then back onto the ambient space:

$$\mathbf{p} = \mathbf{P}\mathbf{P}^\top(\mathbf{w} - \bar{\mathbf{w}}_{\hat{\mathbf{b}}}). \tag{5.15}$$

The residual can then be estimated using the difference between the prediction and the mean-subtracted observation:

$$\mathbf{r} = (\mathbf{w} - \bar{\mathbf{w}}_{\hat{\mathbf{b}}}) - \mathbf{p}. \tag{5.16}$$

This method was validated empirically and found to produce motion predictions of increased accuracy for all trajectories, particularly those far from the original control points $\mathbb{D}_i$.

For the remaining of the thesis, the residuals are the Orthonormal-Basis Residuals and not the 2D Affine residuals used so far, unless explicitly noted.

Evaluating $\mathbf{E}(\mathbf{w}, \mathbf{M})$

The prediction-error scoring function $\mathbf{E}(\mathbf{w}, \mathbf{M})$ is computed as the negative log-likelihood of a zero-mean Normal distribution evaluated at the residuals:

$$E(\mathbf{w}, \mathbf{M}_i) = -\log\left(\mathcal{N}(\mathbf{r}_i, \mathbf{0}, \Sigma)\right). \tag{5.17}$$

The underlying differences between each scoring function will arise from different forms of representing and estimating Σ, which can be generically defined as a, $2F \times 2F$ matrix Σ.

Independent vs. Joint Models

One of the hypothesis we were interested in validating was whether it was possible to obtain useful prediction error scoring functions by considering each $\mathbf{M}_i \in \mathbb{T}$ separately, motivated by computational efficiency, since the use of independent models allows estimating the parameters (in this case Σ) with complexity linear in the number of trajectories (P) and in the number (M) of models in the set $\mathbb{C}$ of candidates: $O(PM)$. In addition, the parameters can be estimated in closed form.

The alternative is estimating the parameters of a joint model, simultaneously considering all the models of motion $\mathbf{M}_i \in \mathbb{T}_j$ for each model combination $\mathbb{T}_j \in \mathbb{S}$. In this case the complexity is linear in the number of trajectories and in the size $\binom{M}{N}$ of the set $\mathbb{S}$ of model combinations: $O(PM^N)$. In addition, joint models (like a Gaussian Mixture Model) require the use of iterative parameter-estimation methods (like Expectation-Maximization), significantly increasing the overall computational cost. The motivation for using this type of joint model, however, is the potential to improve ranking and labeling accuracy.

The next section presents the analysis of the Independent Noise Models (Section 5.3.2), followed by its Joint Noise Model counterpart (Section 5.3.3).

5.3.2 Independent-Noise Likelihood Models

This section explores the ability of three independent likelihood models to obtain useful prediction error scoring functions by considering each $\mathbf{M}$ separately. The independent noise models we explored are:

1. A fixed, constant isotropic $\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$ for all trajectories and all frames.

2. An estimated isotropic covariance $\Sigma_{i,f} = \begin{bmatrix} \sigma_{i,f}^2 & 0 \\ 0 & \sigma_{i,f}^2 \end{bmatrix}$ that fits the residual data of model $\mathbf{M}_i$, at frame f.

3. An estimated 2D covariance $\Sigma_{i,f} = \begin{bmatrix} \sigma_{i,f,xx}^2 & \sigma_{i,f,xy}^2 \\ \sigma_{i,f,yx}^2 & \sigma_{i,f,yy}^2 \end{bmatrix}$ that also fits the residual data of model $\mathbf{M}_i$ at frame f.

Fixed Isotropic σ^2, (*i.e.*, Scaled Euclidean)

The negative log-likelihood of a distribution with a fixed, isotropic σ is equivalent to a Scaled Euclidean distance, plus a constant term:

$$E(\mathbf{w}, \mathbf{M}) = \ln\left((2\pi\sigma)^{\frac{2F}{2}}\right) + \frac{1}{2\sigma^2}\mathbf{r}^\top\mathbf{r} = c + \lambda_s\mathbf{r}^\top\mathbf{r} \tag{5.18}$$

But because $E(\cdot)$ is only used in the context of optimization (Equations 5.9 and 5.11), the constant can be dropped, and the resulting scoring function can be obtained with

$$E(\mathbf{w}, \mathbf{M}) = \lambda_s\mathbf{r}^\top\mathbf{r}, \tag{5.19}$$

where the scale of the isotropic noise becomes a parameter of the model.

While this may be one of the simplest possible prediction-error scoring functions, it was found to work reasonably well in practice. Its primary advantage is computational efficiency, but it also satisfies the first two desirable properties described in Section 5.3 very often. Its failure modes are mainly related to the task of ranking the very best models first.

Estimated, Isotropic σ

As part of the model $\mathbf{M}_n$, the LMMF algorithm returns a vector of isotropic magnitudes for each frame $\sigma_n = [\sigma_1, \sigma_2, \ldots, \sigma_F]$, which for simplicity of

notation can be arranged as

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 & 0 \\ 0 & \sigma_1^2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \sigma_F^2 & 0 \\ 0 & 0 & \dots & 0 & \sigma_F^2 \end{bmatrix}. \tag{5.20}$$

The error scoring function for this model a can then be evaluated using the negative log-likelihood of a regular multivariate normal distribution with

$$E(\mathbf{w}, \mathbf{M}_n) = -\log\left(\mathcal{N}(\mathbf{r}, \mathbf{0}, \Sigma_n)\right). \tag{5.21}$$

The subscripts n indicates that the covariance matrix Σ_n^f is associated to the motion model $\mathbf{M}_n$.

While this model also very fast to evaluate, it was found to very often fail at the task of labeling trajectories given a model combination (as in Equation 5.11), mainly because the estimates of σ^f reflect only the variance of the noise from the inlier subset (from the LMMF algorithm), and not from the whole set of trajectories of object.

Estimated, Non-Isotropic Σ

A full 2D covariance can be used to fit the noise of the orthonormal-basis residuals from the inlier subset of a model $\mathbf{M}$. The resulting 2D covariance matrices can be arranged as

$$\Sigma = \begin{bmatrix} \sigma_{1,xx}^2 & \sigma_{1,xy}^2 & \dots & 0 & 0 \\ \sigma_{1,yx}^2 & \sigma_{1,yy}^2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \sigma_{F,xx}^2 & \sigma_{F,xy}^2 \\ 0 & 0 & \dots & \sigma_{F,yx}^2 & \sigma_{F,yy}^2 \end{bmatrix} \tag{5.22}$$

which can then be used as the noise models for the rest of the trajectories. The error scoring function of this model is:

$$E(\mathbf{w}, \mathbf{M}_n) = -\sum_{f=1}^{F} log\left(\mathcal{N}\left(\mathbf{r}_n^f, \mathbf{0}, \Sigma_n^f\right)\right) \tag{5.23}$$

But while this model is still very fast to evaluate, we observed that additional degrees of freedom make the problem of inaccurate trajectory labeling even worse than with the isotropic version, described above.

5.3.3 Joint-Noise Likelihood Models

Motivated by the need of finding consistently good labelings (from Equation 5.11) we moved from independently estimated models to a jointly estimated one, where both labels and noise models are estimated simultaneously from the data. This model is also beneficial in that it allows fitting of an outlier class, which captures trajectories where tracking failure occurred.

Suppose the motion combination $\mathbb{T} = \{\mathbf{M}_1, \ldots, \mathbf{M}_N\}$ is a set of motions that explains all of the inlier trajectories of the scene, except for the tracking failures (*i.e.*, outlier trajectories). Now, let

$$\mathcal{L}(\mathbb{T}, \mathbf{W}) = \prod_{p=1}^{P}\left(\sum_{n=1}^{N+1} c_n \prod_{f=1}^{F} \mathcal{N}\left(\mathbf{r}_n^f, \mathbf{0}, \Sigma_n^f\right)\right) \tag{5.24}$$

be a Gaussian Mixture Model (GMM) of $N+1$ components, each of which models the zero-mean residual distribution of one of the motion models. The additional Gaussian component models the residuals of the outlier trajectories.

The product over f in Equation 5.24 evaluates the complete likelihood of a trajectory, across all F frames, conditioned on the parameters of the corresponding n^{th} model. The sum incorporates the likelihoods of each of the $N+1$ mixing components, weighted by the mixing proportions c_n, and the product over p estimates the complete likelihood of the entire dataset.

Note that the residual vectors $\mathbf{r}_n^f$ are indexed by n, the motion model index, and are computed with Equation 5.16 using the corresponding motion parameters.

Outlier Model

The selection of a prediction model to compute the outlier model residual vectors was interesting because it had a stronger effect on the performance of the GMM than originally thought. An outlier model that makes too good predictions was found to be competitive with the best estimated motion models, particularly for sequences with trajectories whose motion can be simply modeled (*i.e.* trajectories with constant velocity). This resulted in many false outlier class detections. On the contrary, an outlier model that makes too inaccurate predictions can only be chosen when the inlier model predictions are extremely poor, biasing the model to choose less outliers than potentially necessary.

We found that an outlier model with balanced prediction accuracy could be defined as the mean location of each trajectory across all frames. Using this model, the residual of trajectory $\mathbf{w}$ at frame f corresponds to

$$\mathbf{r}_{N+1}^{f} = \mathbf{w}_f - \frac{1}{F}\sum_{g=1}^{F}\mathbf{w}_g. \tag{5.25}$$

To further regularize the outlier model, and to increase numerical stability when very few outlier trajectories are present, the covariance matrix of the outlier class was restricted to be isotropic, and a small constant was added, as a prior that bounds the smallest possible covariance, artificially limiting the benefit of choosing the outlier class $\left(\Sigma_{N+1} = (\sigma_{N+1}^2)\mathbf{I} + \lambda_\Sigma\right)$.

While it is possible that this model fits stationary objects, the fact that the covariance matrix is limited to isotropic and is regularized by the constant λ_Σ helps prevent the outlier model from explaining a class of static trajectories. In our implementation we used a value of $\lambda_\Sigma = 0.5$.

The parameters (c_n and Σ_n^f) of the GMM are estimated using a typical Expectation-Maximization framework in order to locally maximize Equation 5.24. The ownership probabilities of a trajectory are initialized to zero for all classes except for the one whose model produces the minimum Euclidean distance prediction as indicated by Equation 5.19, which is initialized to one. These ownerships determine the initial mixing proportions and covariance matrices.

Prediction Error

Given the parameters of the GMM estimated in the previous section, the prediction error scoring function for all F frames for each trajectory can be computed using:

$$E(\mathbf{w}, \mathbf{M}_n) = -\log\left(c_n \prod_{f=1}^{F} \mathcal{N}\left(\mathbf{r}_n^f, \mathbf{0}, \Sigma_n^f\right)\right) \tag{5.26}$$

which corresponds to the negative log-likelihood of the trajectory, evaluated using a Gaussian distribution with the estimated covariances and mixing-proportions.

Note that when the GMM is used, the matrix $\mathbf{L}$ of binary labels in the model selection objective function (Equation 5.10) must be augmented to also accommodate for the outlier class $\left(\mathbf{L} \in \{0,1\}^{[P\times(N+1)]}\right)$.

Figure 5.1 shows the labeling that results from fitting the GMM to the residuals of a real sequence. For this example, the model combination $\mathbb{T}$ is the one that optimizes the Euclidean distance metric (of Section 5.3.2) over the set of all possible model combinations $\mathbb{C}$. Color labels correspond to the class with maximum ownership probability:

$$\underset{n\in\{1,..,N+1\}}{\operatorname{argmax}} \; c_n \prod_{f=1}^{F} \mathcal{N}\left(\mathbf{r}_n^f, \mathbf{0}, \Sigma_n^f\right), \tag{5.27}$$

and in this example, the labels qualitatively correspond to the three most salient motions of the sequence: the helmet (that vibrates in front of the camera), the pit crew (that approaches the camera), and the car itself (which remains stationary with respect to the camera).

This sequence was chosen as an illustrative example because it contains a large number of tracking failures (outlier trajectories, shown in black), most of them due to non-rigidly moving reflections, most of which are correctly identified by the GMM outlier class. The log-radial plots to the right correspond to the red, green and blue motion model predictions respectively and these in turn correspond to the car, helmet and pit crew. Note how each model predicts trajectories in its class with smaller residual magnitudes than trajectories on other classes, and how the noise distributions vary widely between trajectories of different classes.

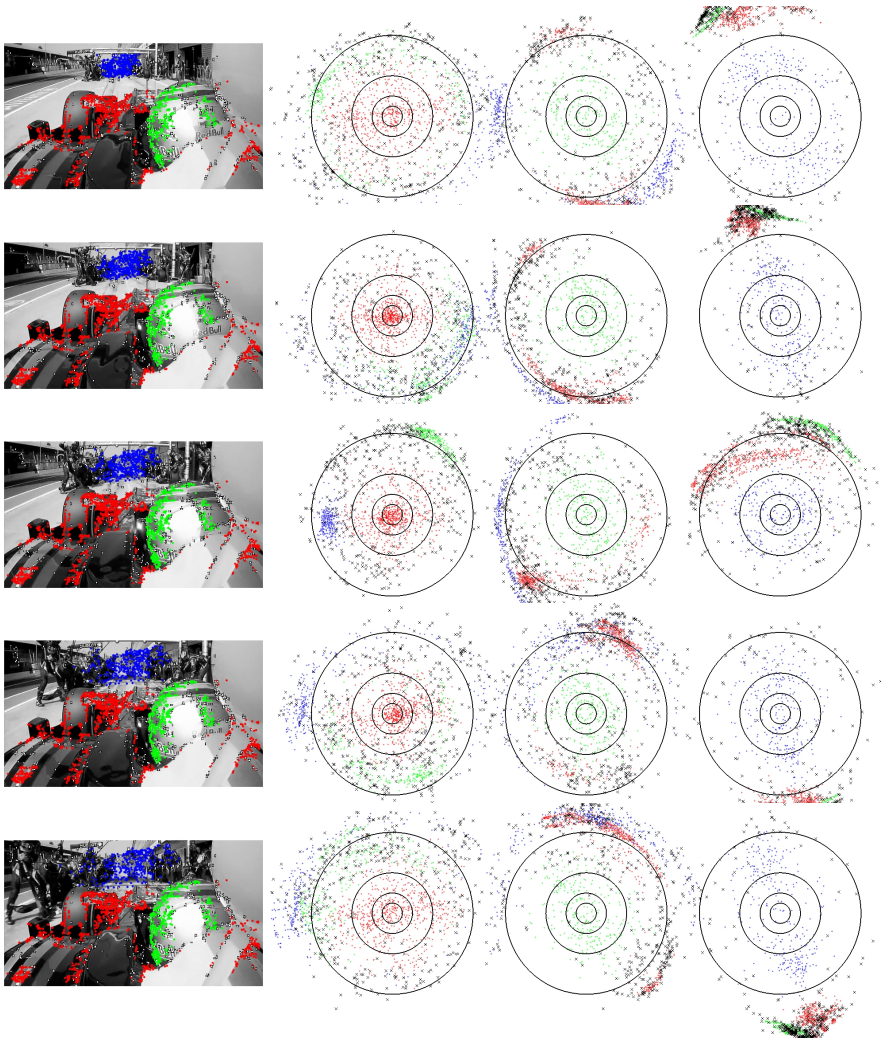


Figure 5.1: Segmentation results for the GMM and the corresponding residuals for 5 frames the `Pitt` sequence. One frame is shown on each row. The red, green and blue labels were obtained using maximum likelihood from the GMM. The log-radial plots show the residuals for all trajectories using each of the models. Outlier trajectories are shown in black.

5.4 Regularization terms

The function $\Phi(\mathbf{w}_p, \mathbf{M}_{j_n})$ in the second term of the model selection objective function (Equation 5.10) penalizes situations where more than one model from $\mathbb{T}_j$ explains the motion of trajectory $\mathbf{w}$ accurately. This term prevents

two undesirable situations. The first one is when one of the offending models is more descriptive (has more parameters) than it should. An example is when a non-degenerate model is used to correctly explain trajectories from a degenerate motion, plus some others from an independently moving object, which are also explained by their own model. Clearly, in this case a simpler model should be used instead. The second undesirable situation is when a model is using its prediction power to explain two independent motions simultaneously, sacrificing accuracy, but then a subset of these trajectories is also explained by one of the other models. These two situations are penalized by the model overlap regularization term. For brevity, let $\hat{E}(\mathbf{w}, \mathbf{M}) = \exp\{-E(\mathbf{w}, \mathbf{M})\}$, then:

$$\Phi(\mathbf{w}, \mathbb{T}_j) = -\log \frac{\max_{\mathbf{M} \in \mathbb{T}_j} \hat{E}(\mathbf{w}, \mathbf{M})}{\sum_{\mathbf{M} \in \mathbb{T}_j} \hat{E}(\mathbf{w}, \mathbf{M})}. \tag{5.28}$$

This Φ is close to zero if trajectory $\mathbf{w}$ has a low error $E(\mathbf{w}, \mathbf{M}_i)$ for just a single model $\mathbf{M}_i$, and high errors for other models. Otherwise, if multiple models $\mathbf{M}_i$ produce similar, nearly minimal errors $E(\mathbf{w}, \mathbf{M}_i)$, then Φ is larger.

The function $\Psi(\mathbf{M}_{j_n})$ in the third term of Equation 5.10 accounts for the number of model parameters for the same reasons as explained in Section 4.3, and is evaluated using Equation 4.43, reproduced here for clarity:

$$\Psi(\mathbf{M}) = \begin{cases} 6(F-1), & \text{if } \omega = 2\text{D} \\ 8(F-1) + I, & \text{if } \omega = 3\text{D}. \end{cases} \tag{5.29}$$

The constants λ_Φ and λ_Ψ modulate the effect of the corresponding regularizer.

5.5 Results

This section presents segmentation accuracy results that motivate some of the design choices. Accuracy is defined as the fraction of individual trajectories with the correct label. For each segmentation and for each sequence, the N computed labels are matched to the N ground truth classes in the optimal way. These results were computed using the `Hopkins 155` dataset, that contains 155 sequences of 2 or 3 moving objects. The dataset contains some articulated motions, as well as many degenerate motions. Some radial

distortion is present, as well as some perspective effects, but otherwise, the tracking noise on these sequences has a relatively small magnitude. Ground truth segmentations are available for the almost 45 thousand trajectories in the dataset. Please note that, while there are a few tracking failures within the `Hopkins 155` dataset, these trajectories are still labeled as belonging to one of the inlier classes. Thus, to prevent the algorithm from arbitrarily losing accuracy by assigning outlier labels, the outlier model is disabled and the most likely inlier class is assigned.

5.5.1 Evaluating the noise model

Sections 5.3.2 and 5.3.3 present a total of four noise models that can be potentially used in the prediction-error scoring function $E(\mathbf{w}, \mathbf{M})$.

The first three make estimates of the noise that are only associated to a single model $\mathbf{M}$ (*i.e.*, are independent to the rest of the models in the combination), with the advantage of computational efficiency. Also, these three noise models are based on the orthonormal-basis residuals (formed by linear projection to the inlier trajectories $\mathbf{W}_{\hat{\mathfrak{b}}}$ only, using Equations 5.15 and 5.16). The first one uses a constant as the magnitude of the isotropic noise model. The second model estimates an isotropic noise model for the residuals, while the third one finds an anisotropic covariance.

The fourth noise model makes joint estimates of the labels and the noise model parameters for all motion models $\mathbf{M}_i$ in the combination $\mathbf{R}$ using a 2D covariance for the noise model (and all the trajectories, not just those in $\mathbf{W}_{\hat{\mathfrak{b}}}$). This results in a characterization of the noise with increased accuracy, but also with significantly increased computational cost. The results from this section are intended to illustrate the effect on segmentation accuracy that results from using each of these models.

Unfortunately, the computational complexity associated with estimating the parameters of the GMM of each model combination in $|\mathbb{S}|$ is prohibitive, and we choose a subset of the best 100 model combinations scored by the objective function $\mathcal{O}_s(\mathbb{T}_j)$ described in Equation 5.10. The parameters of the noise model of these 100 model combinations are then estimated using the GMM approach, as well as the trajectory labels.

Figure 5.2 shows the accuracy results for these four models. Because of the random nature of the algorithm, the experiment was repeated 35 times for

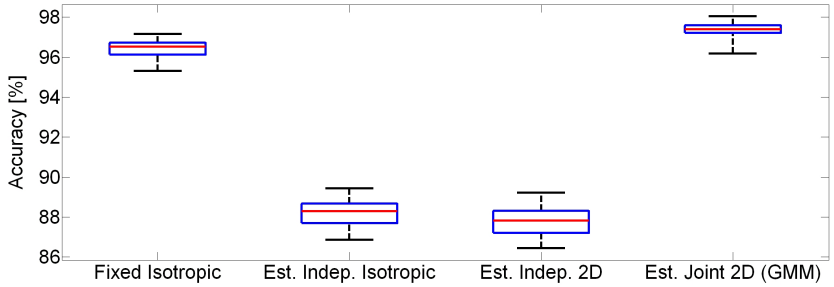


Figure 5.2: Accuracies for the four noise models. The noise models are: isotropic fixed constant, isotropic independently estimated, 2D independently estimated, 2D jointly estimated using a GMM, in that order.

each noise model. Each run consists of four steps: first, execute Algorithm 5.1 to get a list of LMMF models. Second, form the set $\mathbb{S}$ of all model combinations. Third, rank all model combinations using the appropriate noise estimate (for noise models 1 through 3) for the prediction error scoring function $E(\cdot)$ of objective function $\mathcal{O}_s(\cdot)$. For the remaining noise model (the one estimated using a GMM), the highest ranking model combinations produced using the fixed isotropic noise model are ranked a second time, using the GMM joint noise model. And fourth, pass the resulting labelings and objective function evaluations to a refinement stage (next Chapter) that estimates the final segmentation result.

The plot suggests that the simple isotropic model has good performance, and that this performance can be improved by subsequently fitting mixture models to the top ranked results. The second and third methods appear to suffer from poor noise estimates obtained from the restricted set of inliers of each of the LMMF models (as discussed above).

5.5.2 Sub-sampling the Set $\mathbb{S}$

In the interest of computational efficiency, we questioned the need of exploring the entire set of possible motion combinations $\mathbb{S}$ to find a good solution to Equation 5.9. In other words, we wanted to know how much segmentation accuracy is lost when one explores only a fraction of the combinations in $\mathbb{S}$ while looking for an approximation to the optimal. The results of this experiment are shown in Figure 5.3. For each box-plot, a fraction $\lambda_{\mathbb{S}}$ of

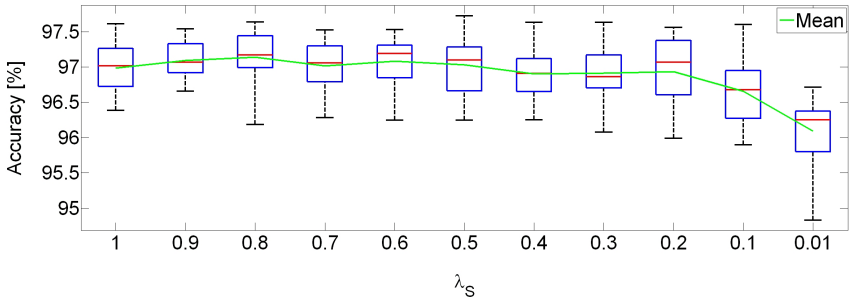


Figure 5.3: Accuracy box-plot for experiments with subsets of model combinations of different sizes. The optimization of Equation 5.9 is limited to a subset of $\lambda_{\mathbb{S}}|\mathbb{S}|$ randomly chosen models from $\mathbb{S}$. The plot for $\lambda_{\mathbb{S}} = 1$ is shown as a baseline.

randomly chosen, uniformly-distributed, model combinations from $\mathbb{M}$ was used. The range $\lambda_{\mathbb{M}} \in \{1.0, 0.9, \ldots, 0.1, 0.01\}$ was explored. Each box-plot is constructed with the results of 20 runs of the algorithm on the standard `Hopkins 155` dataset. Overall accuracy did not change significantly in the range $1 > \lambda_{\mathbb{M}} > 0.2$, which translates to almost an entire order of magnitude in computation time savings for the model-selection stage of the algorithm.

This experiment also indicates that there is a large amount of redundancy between model combinations $\mathbb{T}_i$ and between models themselves $\mathbf{M}_j$. This is not very surprising as in most cases, the Local Motion Model Fitting will produce a plausible hypothesis for a motion model, and in order to explain the motion of a sequence with relatively good accuracy, the only requirement is that models do not overlap.

Finally, note that the set $\mathbb{S}$ in this plot has either $\binom{70}{3} = 328,440$ for sequences with 3 motions, or $\binom{70}{2} = 4,830$ for sequences with two motions. When these numbers are multiplied by the smallest $\lambda_{\mathbb{S}} = 0.01$ the algorithm is left with 3280 and 50 model combinations, respectively, yet the resulting average accuracy loss is less than 1%.

Other datasets with sequences where the number of independent motions is larger ($N = 5$) are consistently and correctly motion segmented with very limited subsets of $\mathbb{S}$. For example, in Chapter 7 we show good results for $N = 5$ using just 10^5 of all possible combinations (an implicit $\lambda_{\mathbb{S}} = 6.88 \times 10^{-6}$). Nevertheless, more efficient sampling schemes (besides uniform sampling) are an interesting avenue for future research, especially for sequences with more than 5 rigid motions.

6 Model Averaging

This chapter describes a mechanism that improves the overall segmentation accuracy of the algorithm by using intermediate results from the MMMF of Chapter 5 alone. The approach was also found to increase the repeatability of the labelings across multiple runs of the algorithm.

The motivation to do model averaging comes from the observation regarding the presence of noise in the MS labels associated to the optimal model combination. This manifests as labeling errors given to a few trajectories. We hypothesize that this noise is introduced by the local nature of the estimated motion models.

Two key observations suggest that it should be possible to reduce the effects of this type of noise. First, the labelings from the close-to-optimal model combinations are often just as accurate, if not more, than the labeling associated to the optimal model combination. This is possible because different motion combinations are often built from plausible and diverse sets of model hypotheses, rendering accurate predictions for all of the trajectories. And second, the noise that contaminates these labellings is often independent across different model combinations, as many of these labelings arise from independently instantiated local motion models.

We thus hypothesize that a model averaging approach is likely to reduce the effects of this local type of noise, potentially improving segmentation accuracy and reducing result variability. The rest of this chapter provides evidence of the above observations, explains how the model averaging procedure is implemented and concludes with experiments that confirm the hypothesis of improved accuracy and reduced result variability.

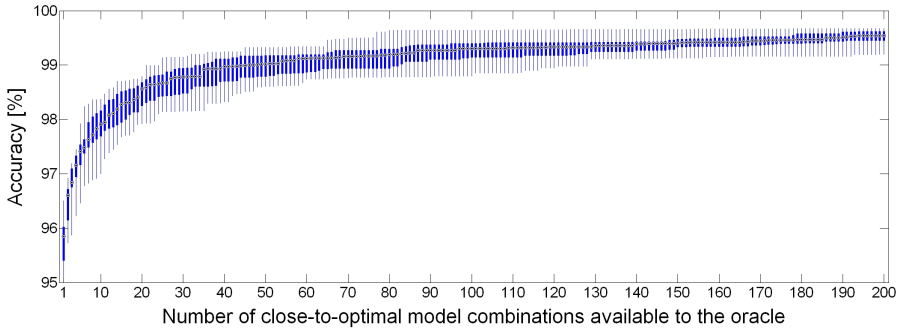


Figure 6.1: Box-plot of accuracy on the `Hopkins155` dataset. An oracle identifies the most accurate segmentation result from within the set of t closer-to-optimal model combinations for each sequence. Results with 99% accuracy are available within the first $t = 50$ model combinations. This box-plot is constructed with 200 runs of the algorithm.

6.1 Motivation

Empirical evidence that supports the above observations can be found in Figures 6.1, 6.2 and 6.3. Figure 6.1 shows a box-plot of overall accuracy (percentage of correctly classified trajectories) from the `Hopkins 155` dataset. In this plot an oracle that uses ground truth identifies the most accurate segmentation labeling from within the set of t closer-to-optimal model combinations for each sequence. The value of t varies in the range $t \in [1, 200]$ along the horizontal axis. The plot answers the question: if we could tell the best labeling from within the pool of t closer-to-optimal labelings, what would the overall accuracy be? As it happens, the resulting accuracy for $t = 50$ is better than the state of the art by a large margin (and for $t = 200$ the overall error is almost three times smaller than the state of the art), suggesting that, besides the information provided by the optimal labeling (whose corresponding accuracy is shown at $t = 1$), close-to-optimal labelings also contain relevant information which may be used to improve segmentation performance.

The plot of Figure 6.2 shows the relative frequency with which the MMMF objective function places the most accurate combination within the best 200 slots. Clearly, there is a preference to rank the best model combination first, but this plot also suggests that the most accurate labeling is not in the first few closest-to-optimal model combinations fairly frequently.

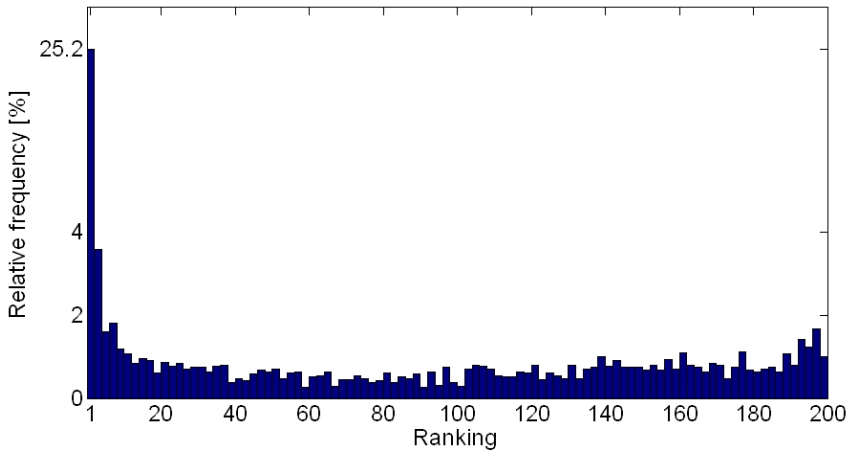


Figure 6.2: Histogram of the ranking of the best model (as determined from ground truth), when the ranking is determined using the Model Selection Objective function.

Finally, Figure 6.3 shows a grid with 16 labelings generated by some of the best 200 motion model combinations from a typical `Hopkins 155` sequence. Color coding indicates the resulting segmentation for each case. Bigger points indicate the model inliers ($\tilde{\mathbf{b}}$, which are used to estimate the model orthonormal basis $\mathbf{P}$) for each of the 3 models of motion. These plots illustrate how different labelings have small, and probably independent deviations from the correct labelling. Our independence hypothesis is supported by the fact that motion models are also independently instantiated.

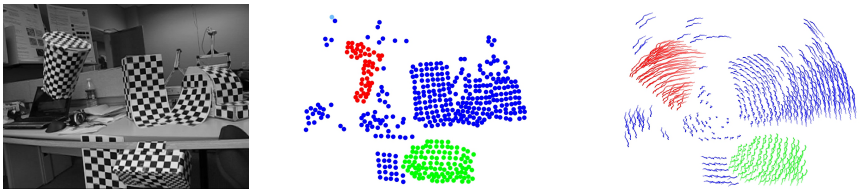
6.2 Formulation

T Number of closer-to-optimal labelings to be averaged.

If the above hypotheses are true, the goal at this stage is to average the resulting labelings of the T closer-to-optimal motion combinations $\mathcal{T}$ into what should be an improved final labeling. For that purpose, let

$$\hat{\mathbb{C}} = [\mathbb{T}_1^\star, \mathbb{T}_2^\star, \ldots, \mathbb{T}_T^\star] \tag{6.1}$$

with $\mathbb{T}_t^\star \in \mathbb{S}$ and $\mathcal{O}_S(\mathbb{T}_t^\star) \leq \mathcal{O}_S(\mathbb{T}_{t+1}^\star)$, be the T-tuple of sorted, highest ranking model combinations using the score given by the model selection objective function (of Equation 5.10).



Original image (left). Ground truth color labeling of trajectories at frame 1 (center), and across all frames (right) of sequence `1R2RCR` from the `Hopkins 155` dataset.

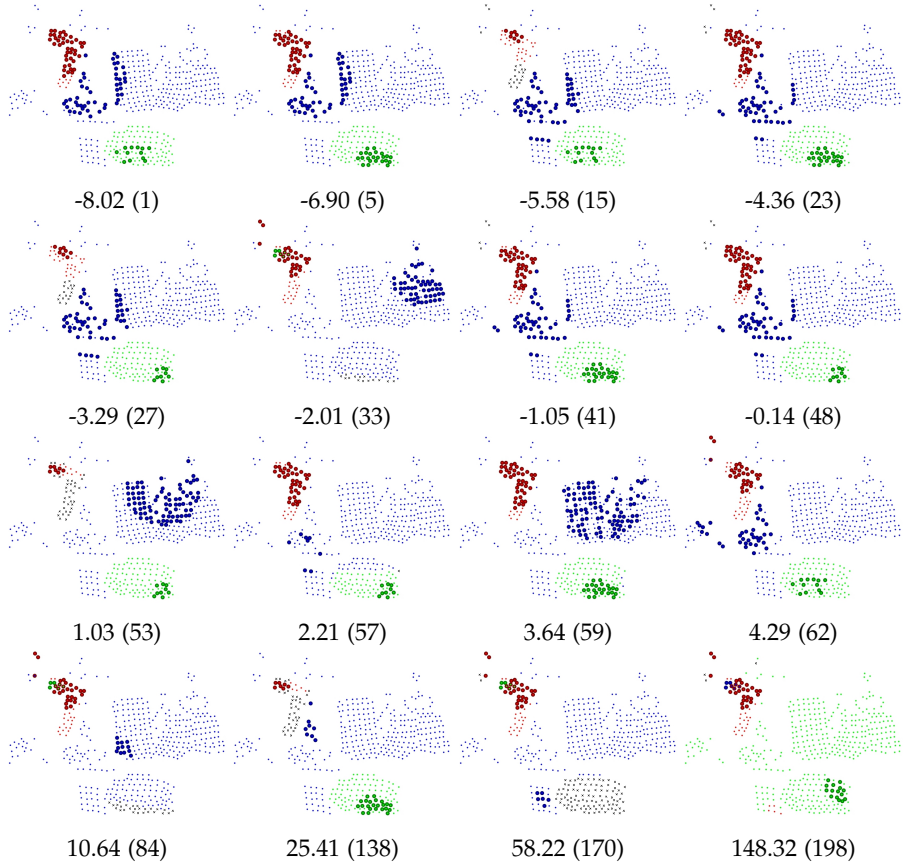


Figure 6.3: Labelings for several motion model combinations. Motion model control points ($\tilde{\mathbf{b}}$, used to estimate the model orthonormal basis $\mathbf{P}$) are shown as big dots. Resulting labeling as color coded smaller dots. Captions for each figure indicate the objective function evaluation (and its corresponding overall rank).

Now, using the deterministic mapping $\mathbb{T} \xrightarrow{\mathcal{F}} \mathbf{L}$ that takes model combina-

tions ($\mathbb{T}$) and returns trajectory labels ($\mathbf{L}$) from Equation 5.12, let

$$\mathbb{L} = [\mathbf{L}_1, \mathbf{L}_2, \ldots, \mathbf{L}_T] \tag{6.2}$$

with $\mathbf{L}_t = \mathcal{F}(\mathbb{T}_t)$ denoting the T-tuple of labelings that corresponds to the t^{th} ranked model combination $\mathbb{T}_t$.

Keep in mind that because this result originates from the `Hopkins 155` dataset, the outlier class was disabled and the labellings are matrices $\mathbf{L} \in \{0,1\}^{I \times N}$ (as opposed to the more general case that includes an outlier class and the labelling reflects so $\mathbf{L} \in \{0,1\}^{I \times (N+1)}$). This implicitly means that Equation 5.10 returns the most likely inlier class (and never the outlier class). This is true all throughout this chapter.

6.3 Affinity Matrix

Result averaging is achieved by first computing an affinity matrix $\mathbf{Z}$ for all pairs of trajectories. This matrix captures the frequency with which pairs of trajectories share the same label in each $\mathbf{L}_t \in \mathbb{L}$. Then, an off-the-shelf spectral clustering algorithm [34] is run on the resulting affinity matrix to obtain the final MS result.

In the general context of clustering, the entry $z_{i,j}$ of an affinity matrix $\mathbf{Z}$ is a metric of similarity between the i^{th} and the j^{th} data points. In our context, similarity between pairs of trajectories is defined as the frequency with which trajectory i has the same label as trajectory j within the labelings in the set $\mathbb{L}$.

For this purpose, let $f_{\mathbb{L}}(i, j, \mathbf{L}_t)$ be a function that equals one when the labels of the i^{th} and j^{th} trajectories are the same, according to the labeling $\mathbf{L}_t$, and zero otherwise:

$$f_{\mathbb{L}}(i, j, \mathbf{L}_t) = \begin{cases} 1, & \mathbf{l}_t^i = \mathbf{l}_t^j \\ 0, & \text{otherwise.} \end{cases} \tag{6.3}$$

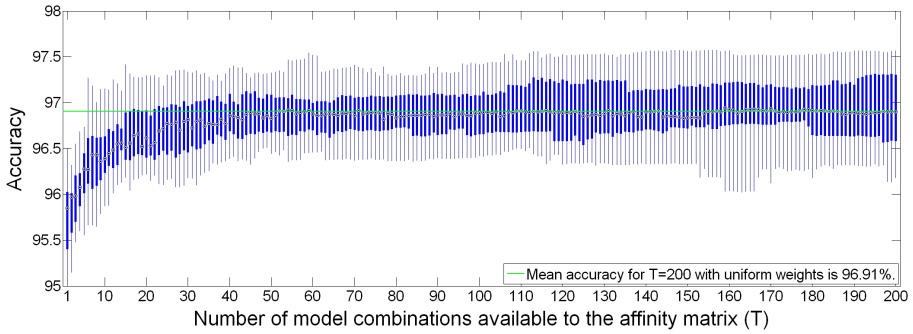


Figure 6.4: Box-plot of overall accuracy on the `Hopkins 155` dataset when averaging up to T labelings. The entry at $T = 1$ shows the accuracy of the the optimal model combination (with no averaging).

6.3.1 Model Averaging

Using Equation 6.3, the entry $z_{i,j}$ of the affinity matrix can be computed with a simple average, as in:

$$z_{i,j} = \frac{1}{T} \sum_{t=1}^{T} f_{\mathbb{L}}(i,\, j,\, \mathbf{L}_t). \tag{6.4}$$

We found that the averaging procedure does result in improved overall accuracy of the MS labeling, even for small values of T. Figure 6.4 shows overall accuracy box-plots over the entire `Hopkins 155` dataset for values of $T \in [2,\ 200]$. Each box-plot contains data from 20 runs of the algorithm. The labeling that corresponds to the optimal motion combination alone (without any averaging) is shown as a baseline, at $T = 1$.

6.3.2 Non-uniformly Weighted Model Averaging

Despite the improvement, a model that simply averages all the labelings together fails to acknowledge the relative reliability across different labelings. In fact, motion combinations that result in a closer to optimal objective function evaluation, often also produce more accurate labelings. Empirical evidence that supports this idea can be found in Figure 6.5. The plot shows accuracy box-plots for segmentation results over the entire `Hopkins 155` dataset when using only the t^{th} model combination, sorted by the model selection objec-

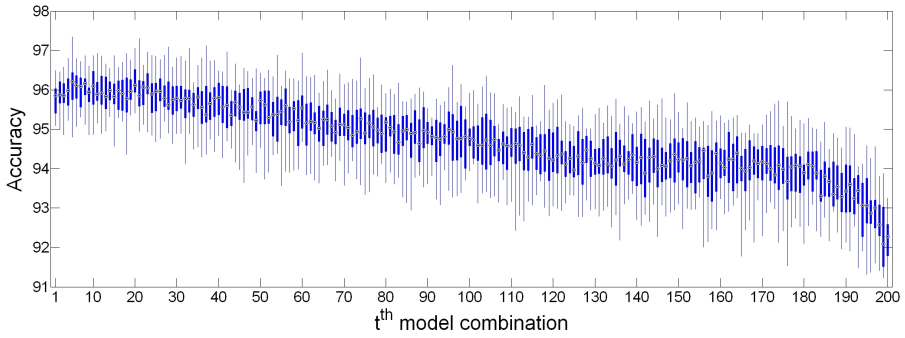


Figure 6.5: Accuracy of the t^{th} model combination, sorted by the model selection objective function.

tive function, to produce the output labeling (without averaging). This plot suggests that closer-to-optimal model combinations typically produce more accurate segmentations.

To incorporate this knowledge, Equation 6.4 is generalized to non-uniformly weight the contribution of each labeling:

$$z_{i,j} = \sum_{t=1}^{T} w_t f_{\mathbb{L}}(i, j, \mathcal{L}_t), \tag{6.5}$$

and the weights w_t can be computed as a function of the model selection objective function scores:

$$w_t = \frac{\exp\{-\lambda_{\mathbb{L}} \mathcal{O}_s(\mathcal{M}_t)\}}{\sum_{t=1}^{T} \exp\{-\lambda_{\mathbb{L}} \mathcal{O}_s(\mathcal{M}_t)\}}. \tag{6.6}$$

The denominator normalizes the weights so that $\sum_t w_t = 1$, and the parameter $\lambda_{\mathbb{L}}$ governs the dispersion of weights. A very small value for $\lambda_{\mathbb{L}}$ leads to a weight distribution where the relative differences between objective function evaluations become irrelevant, and the weights become uniform:

$$\lim_{\lambda_{\mathbb{L}} \to 0} w_t = \frac{1}{T}. \tag{6.7}$$

In contrast, giving $\lambda_{\mathbb{L}}$ a large value reduces the model averaging to simply

using the labeling from the optimal model combination alone, since:

$$\lim_{\lambda_{\mathbb{L}} \to \infty} w_t = \begin{cases} 1, & t = 1 \\ 0, & \text{otherwise,} \end{cases} \tag{6.8}$$

suggesting that the parameter could be empirically tuned to maximize accuracy.

The effect of different values of $\lambda_{\mathbb{L}} = [0,\ 0.001,\ 0.01,\ 0.1,\ 0.25,\ 0.5,\ 1.0]$ was studied as the number of model combinations varied in $T \in [2, 200]$. The results revealed how, as the value of $\lambda_{\mathbb{L}}$ increases steeply, the overall accuracy was restricted in a similar way as if only a few labelings were being averaged in, confirming the notion explained above. Figure 6.6 shows how the accuracy improvement is truncated earlier every time a larger value of $\lambda_{\mathbb{L}}$ is used.

Results for mean accuracy across 20 runs of the algorithm on the `Hopkins 155` dataset for values of $\lambda_{\mathbb{L}} \in \{0,\ 0.001,\ 0.01,\ 0.1,\ 0.25,\ 0.5,\ 1.0\}$, for $T \in [2, 200]$ are shown in Figure 6.7. This plot suggests that the value of $\lambda_{\mathbb{L}} = 0.01$ consistently leads to increased accuracy with respect to the labeling obtained with uniform weights ($\lambda_{\mathbb{L}} = 0$), across all values of T, although by a small margin. As with previous figures, the value at $T = 1$ shows the labeling from the optimal model combination, with no averaging (shown as a baseline).

A summary of the results at $T = 200$ is shown in Figure 6.8 for different values of $\lambda_{\mathbb{L}}$. The "No averaging" baseline corresponds to the accuracy of the optimal labeling identified by the best score from the model selection objective function.

In order to quantitatively evaluate result variability, we estimated the standard deviation of the accuracy of the computed segmentation for each sequence across the 20 runs of the algorithm for each value of T. The resulting standard deviations were then weighted-averaged according to the number of trajectories in each sequence to obtain a metric of inter-sequence result variability, averaged over the entire `Hopkins 155` dataset. The resulting variances are shown in Figure 6.9. The plot suggests that result variability is also reduced by a large margin for values of $\lambda_{\mathbb{L}} < 1$, compared to the non-averaged baseline.

Finally, with the goal of better understanding where model averaging finds the improvements, we looked at per-sequence accuracy statistics. The plot in Figure 6.10 shows accuracy box-plots for each sequence, sorted by median

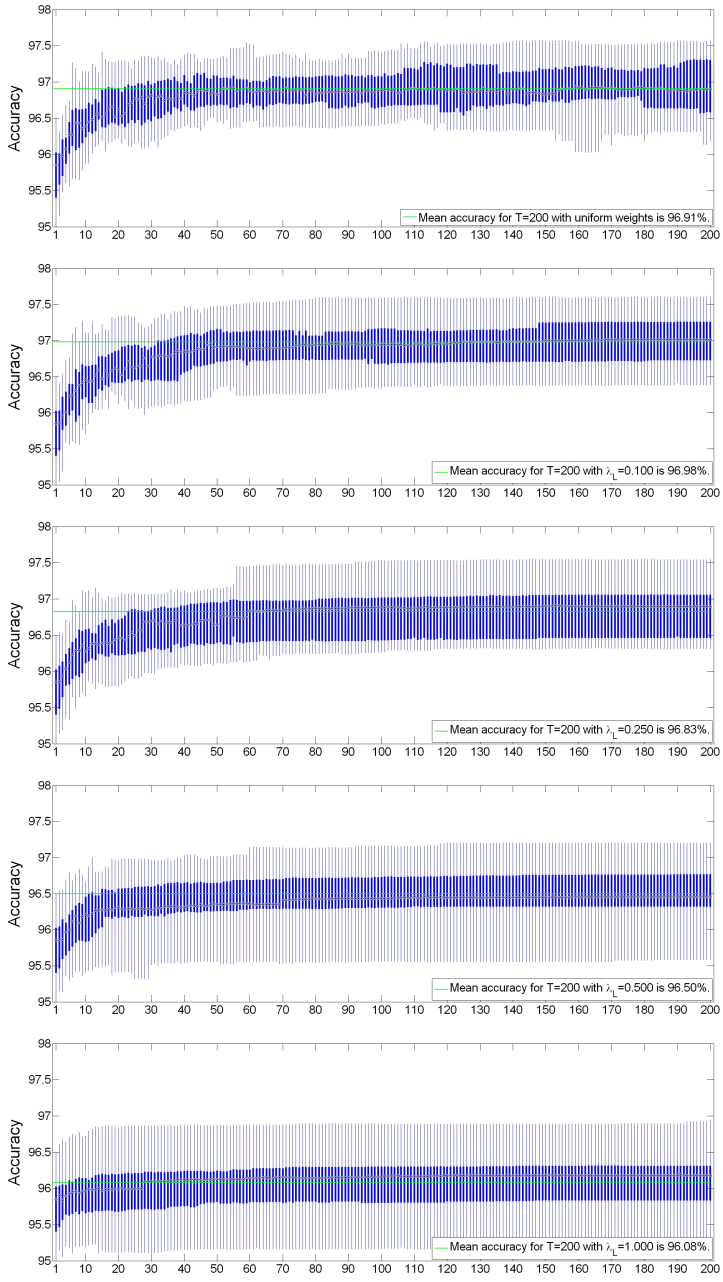


Figure 6.6: Box-plots of overall accuracy for $\lambda_{\mathbb{L}} = \{0.0,\ 0.1,\ 0.25,\ 0.5,\ 1.0\}$.

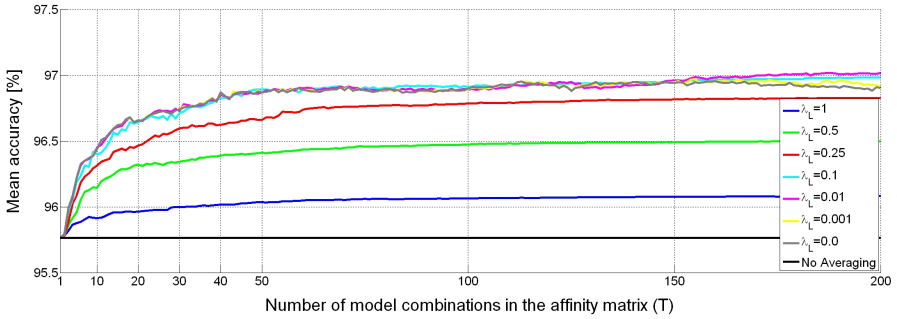


Figure 6.7: Mean accuracy of the t^{th} model combination for different values of $\lambda_{\mathbb{L}}$.

accuracy (Sequence indices in $[1, 155]$ are shown in the x axis). The box-plot at the top is the baseline, computed with the labeling that corresponded to the optimal model subset (without averaging). The rest of the plots are averaged results using values of $\lambda_{\mathbb{L}} = \{0.1,\ 0.25,\ 1.0\}$, from top to bottom, respectively. These plots suggest that only a few sequences result in consistent mis-classification errors, and when compared to the baseline, it is also clear that model averaging does increase the segmentation accuracy of some of the problematic sequences and also reduces result variability.

6.4 Definitive Pipeline

The pipeline that was used to compute our most competitive results is as follows:

- Instantiate $\lambda_{\mathbb{C}}$ (typically $\lambda_{\mathbb{C}} = 70$) models of motion using the LMMF algorithm (of Chapter 4) to populate the set $\mathbb{C}$, as explained in Chapter 5.

- Randomly choose a subset of $\lambda_{\mathbb{T}}$ (typically $\lambda_{\mathbb{T}} = 10,000$) model combinations from the set $\mathbb{S}$ and rank them using the Euclidean distance version (Equation 5.19) of the Model-Selection evaluation function $\mathcal{O}_S(\cdot)$.

- Find the subset of the λ_{GMM} (typically $\lambda_{GMM} = 100$) best scoring models from the previous step and now score them using the GMM version of the model-selection evaluation function.

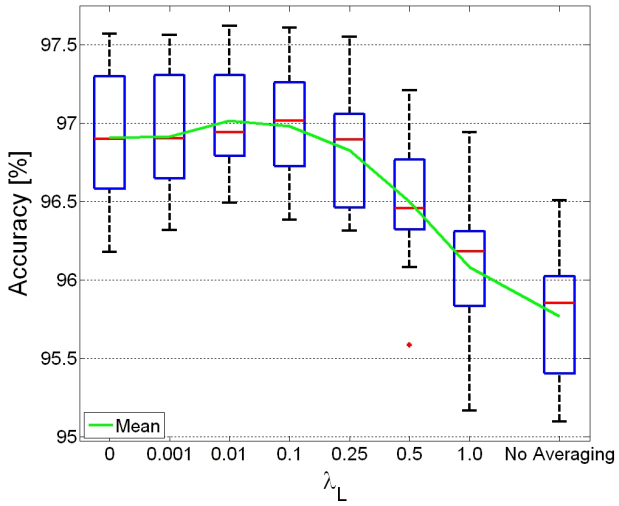


Figure 6.8: Accuracy with $T = 200$ for different values of $\lambda_{\mathbb{L}}$. Results are compared against a "No Averaging" baseline.

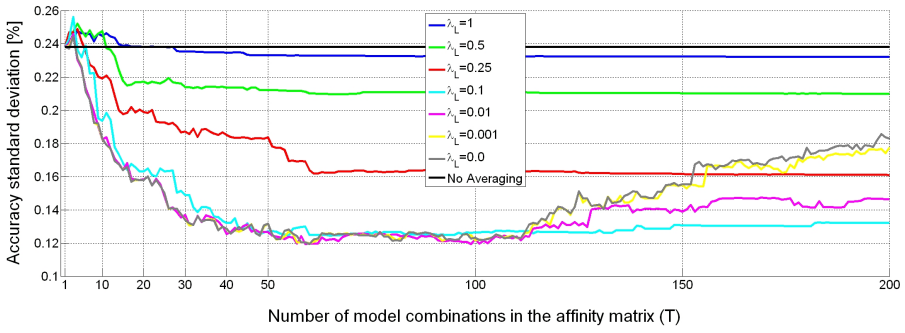


Figure 6.9: Metric of result variation for the t^{th} model combination for different values of $\lambda_{\mathbb{L}}$. See text for details.

Table 6.1: Mean absolute error and accuracies for the estimated number of models. Our method outperforms all prior art by a significant margin.

Method	Average Accuracy [%]	Computation Time [s]
SSC [10]	98.76	14,500
CPA [61]	98.75	147,600
RANSAC	89.15	30
Ours	97.05	400

- Perform modeling averaging using the resulting labelings and objective function evaluations from the previous step.
- (Optional) Using the resulting the inlier labels for each class, estimate motion models (as orthonormal basis) using the Matrix Factorization technique.

With this settings our method obtains a mean accuracy of 97.05% on the `Hopkins 155` dataset. A summary of accuracies and run-times for current state of the art algorithms is shown in Table 6.1. The important aspect to notice is that our method is between 2 and 3 orders of magnitude faster with only a small accuracy loss.

6.5 Conclusions

Results presented throughout this chapter (particularly Figures 6.7 and 6.8) strongly suggest that averaging labelings from independently instantiated model combinations improves segmentation accuracy.

Further improvements (although potentially marginal), were obtained by non-uniformly weighting the contribution of each labeling according to a function of the model-selection evaluation score, although this requires estimating the parameter $\lambda_{\mathbb{L}}$. The alternative is always using $\lambda_{\mathbb{L}} = 0$, but in this case, an appropriate value for T must be found in order to prevent the inclusion of too many low-quality labelings that may contaminate the averaged labeling result (note that the accuracy plot for constant weights $\lambda_{\mathbb{L}} = 0$ decreases after $T = 150$).

The analysis on this chapter also provides some insight regarding the earlier sections of the algorithm. The box-plot of Figure 6.1 indicates that accuracies of over 99% can be reached using the best $T = 50$ subsets of the

motion models. This suggests that the model fitting algorithm (of Chapter 4), is capable of instantiating plausible and accurate motion model hypotheses for the overwhelming majority of the independently moving objects in each scene. And that the model selection objective function is capable of ranking the best model combinations within the first few (50) candidates. It is a matter of further investigation whether identifying the correct solution (from a set of 50 plausible solutions) is an easier problem than improving on the MS algorithm itself.

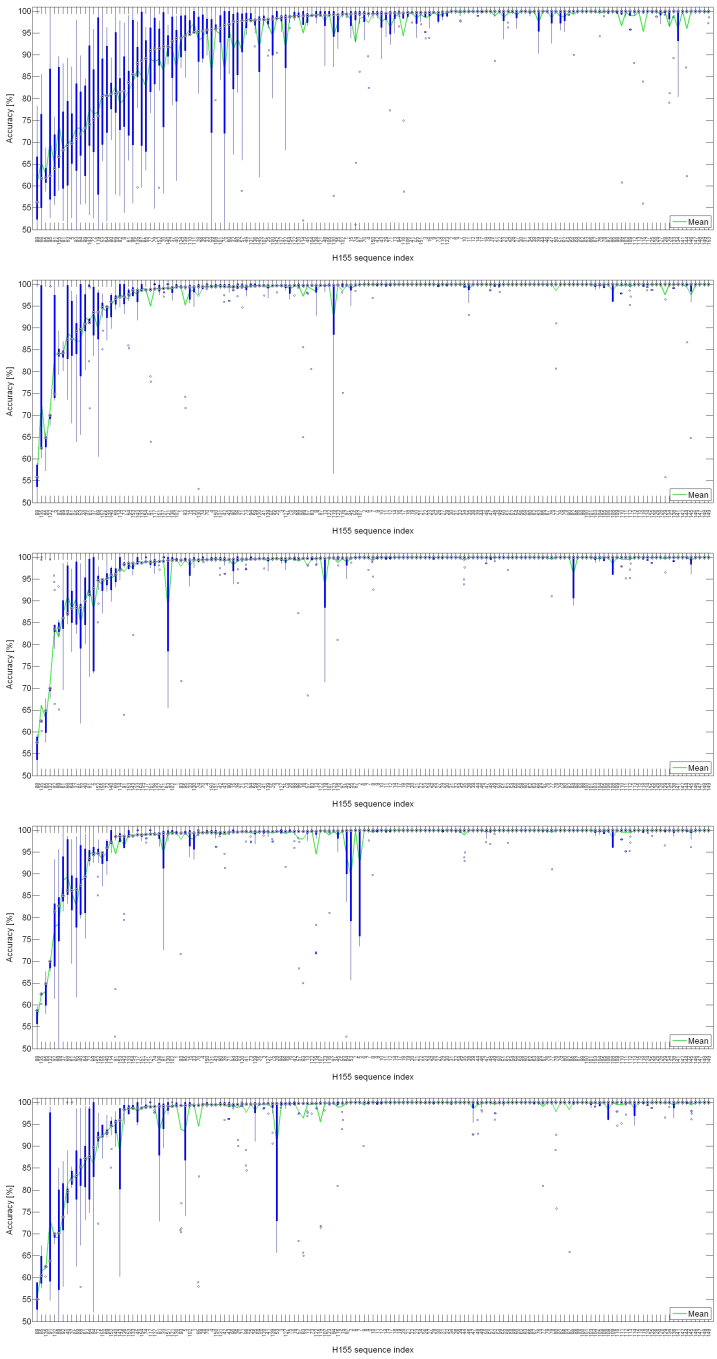


Figure 6.10: Box-plots of accuracy per sequence. Top row: No-averaging. Rows 2-5: Averaging with $\lambda_{\mathbb{L}} = \{0.1,\ 0.25,\ 0.5,\ 1.0\}$ respectively. Sequences are sorted by median accuracy for display purposes.

7 Other Results

This chapter presents results from experiments that test the overall performance of the algorithm under less conventional and previously untested conditions. These include experiments that quantitatively evaluate robustness to noise, or that qualitatively characterize the effects of incorrectly specifying the number of motions in the scene. We also test the algorithm on datasets that do not conform to the standard working assumptions, like rigid-motion or orthographic projection, and draw conclusions on the observed performance.

Unless noted otherwise, all the segmentation results presented in this chapter are computed with the same set of parameters.

The final section of this chapter includes an extension to the original algorithm that enables estimation of the number of independent motions in the scene with state of the art results.

7.1 Unstructured Noise of Varying Magnitude

Noise is present in all trajectory data and within the context of orthographic rigid motion segmentation, even unaccounted image formation phenomena like perspective effects or lens distortion can contribute to model deviations that can be generally be referred to as noise. The goal of this section is to quantitatively evaluate the effects of unstructured noise of different magnitudes on the overall segmentation accuracy, and to compare the results with the state of the art.

To isolate the effects of magnitude-controlled, zero-mean normally-distributed noise from the inherent real noise of a non-synthetic sequence, we start

by constructing a noiseless version of the `Hopkins 155` dataset. The noiseless version of each sequence is computed using a rank-limited reconstruction of the mean-subtracted trajectories of each ground truth segment. Because the complexity of each motion is unknown (degenerate or non-degenerate), we conservatively choose to compute rank 3 reconstructions for the trajectories of each motion segment. This rank limited reconstruction is considered the affine noiseless version of the dataset. Then, multiple versions of the entire `Hopkins 155` dataset are generated by adding normally-distributed, zero-mean noise to the noiseless version. Each noise-controlled version has an associated magnitude σ_n with $\sigma_n \in \{0.01, 0.25, 0.5, 1, 2, 4\}$.

Figure 7.1 shows the average accuracy that results from running Sparse Subspace Clustering [10] and our method on the noise-controlled data. Both algorithms were run 20 times. The error bars indicate one standard deviation of result variability. The graph suggests that our method compromises accuracy only for large levels of noise, and does comparatively good or better for values of $\sigma < 1.0$, while still being computationally much more efficient.

Note that the synthetic noise results shown in figure 7.1 do not appear to be simultaneously consistent with both the SSC and our algorithm's performance at any specific noise magnitude σ_n. We conjecture that this is due to unaccounted-for image formation processes that result in structured noise, such as barrel distortion (visible in some sequences) or perspective effects. An alternative hypothesis is that these synthetic sequences are biased in favor of our method, given that the synthetic noise dataset was built from rank limited initial data and contaminated with normally distributed noise. Because our method relies on matrix factorization which optimizes the reconstruction of the data a least squares sense, normally distributed noise is favorable.

The segmentation accuracy of both methods stays over 80% even for relatively high noise values (*i.e.*, $\sigma_n \geq 8.0$), however our method's accuracy drops faster as the magnitude of the noise increases. This could be a consequence of the reliance on the extrapolation of spatially local motion models, which can be expected to deteriorate with increasing levels of noise.

7.2 Broken Assumptions

The vast majority of motion segmentation methods assume that the number of motions is known and that the underlying motions are rigid. Ours does too.

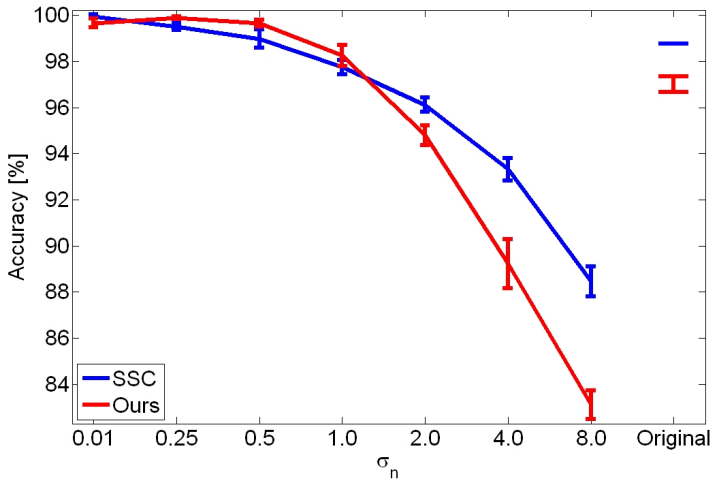


Figure 7.1: Average accuracy across artificial `Hopkins 155` datasets with controlled levels of zero-mean Gaussian noise. The accuracy on the original `Hopkins 155` dataset is shown in the right-most entry of the plot (error bars for SSC are not shown because result variability is not available for this method).

This section challenges these and other underlying assumptions and presents results of our method using datasets or input parameters that deviate from them.

7.2.1 Incorrect Motion Count

An obvious drawback of most motion segmentation algorithms is the need to indicate the number of motions N in the scene, and while some methods to estimate this number already exist (see Section 7.3 for a few examples of existing methods and a new proposal), the difficulty of the problem and the ambiguous nature of the data still leads to imperfect results. Even when a human operator determines the number of motions, the call is influenced by the same priors that enable other perceptual grouping tasks, like local coherence, contrast saliency or semantic saliency, which also leads to inaccurate estimates (as will be shown below). In fact, the matter can quickly become philosophical when one wonders about the definition of the "correct" number of motions in a scene, although we tend to align with the idea of this number being the minimum necessary to recover the underlying 3D structure of the scene [24].

Still, assuming the true number of independent motions N is obvious

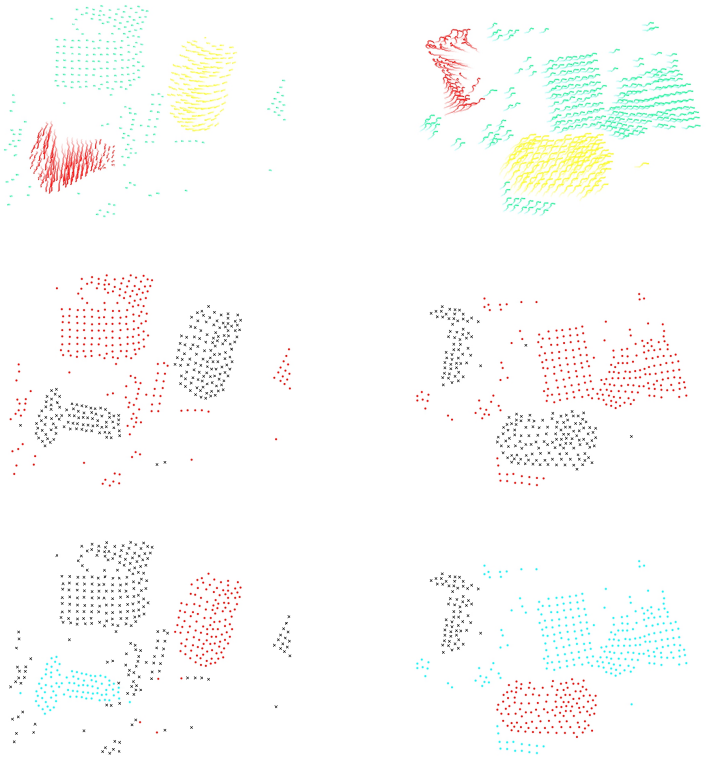


Figure 7.2: Two sequences (one left, one right) from the `Hopkins 155` dataset with little noise, highly distinct independent motions and $N = 3$. Top row: ground-truth color coded trajectories. Rows 2-3: Segmentation results with $N_I = 1$ and 2 respectively. Colored dots correspond to estimated inlier labelings, black crosses labels for outlier trajectories.

from visual inspection, the goal is to qualitatively evaluate how the proposed algorithm behaves when the input number of motions N_I is different from N. Several rigid, as well as some semi-rigid sequences are used for the analysis. If a sequence has N independent motions, a set of hypothesis for the numbers of independent motions that includes at least $N_I \in \{N-1, N, N+1\}$ is tested, the results visually inspected and some conclusions drawn. The rest of this section describes results that are representative of some of the most typical outcomes for both under-estimations ($N_I < N$), as well as over-estimations ($N_I > N$). Each subsection talks about different types of sequences.

Rigid sequences with Salient Motions and Noise of Small Magnitude

The cases presented here come from the standard `Hopkins 155` dataset and in these first few examples, the motions are very distinct. Under- and over-segmentation results for 2 sequences with three independently moving objects ($N = 3$) are shown in Figures 7.2 and 7.3, respectively. As with most Figures in this section, the top image corresponds to color-coded trajectories using ground truth (or when GT is unavailable, a qualitatively correct motion segmentation labeling is used instead). Subsequent images show segmentation results with increasing values of N_I. Please note that the outlier model is enabled for all the experiments reported in this chapter.

The results of figure 7.2 suggest that when N_I is under-estimated, the outlier model is utilized to group the trajectories from objects whose motion was not explained by any of the inlier motions. We see this as a desirable feature. It can also be observed that the inlier model(s) explain the cluster(s) with the smallest residual magnitudes. This behavior responds to the nature of the objective function that is being optimized, where models with better prediction accuracy are preferred, regardless of the number of model-inlier trajectories. We also think this behavior is beneficial, for instance when the number of motions is well under-estimated (as in Figure 7.2, row 2), allowing the one and only inlier cluster to explain the motion of a real independently moving object, instead of other alternatives where more trajectories could be explained, albeit less accurately.

Figure 7.3 shows results for cases when N_I is over-estimated. The algorithm's behavior is much less predictable, and while in some cases the resulting over-segmentation separates true clusters into perceptually relevant components (like the separation of all objects into its planar components, as shown in the left column for $N_I = 5$), it is also true that casual co-occurrence of some type of noise (like unaccounted perspective effects) in a group of trajectories may be sufficient to drive the usage of a whole segment without a semantically relevant meaning, like in the right column example when $N_I = 5$ (Figure 7.3, right column, bottom row) where the pink segment corresponds to trajectories that are not well explained by the green labeled-motion model, and are independently modeled thanks to the availability of the extra motion model. It may be important to note, however, that the union of the correct subsets still renders the correct segmentation obtained when $N = 3$.

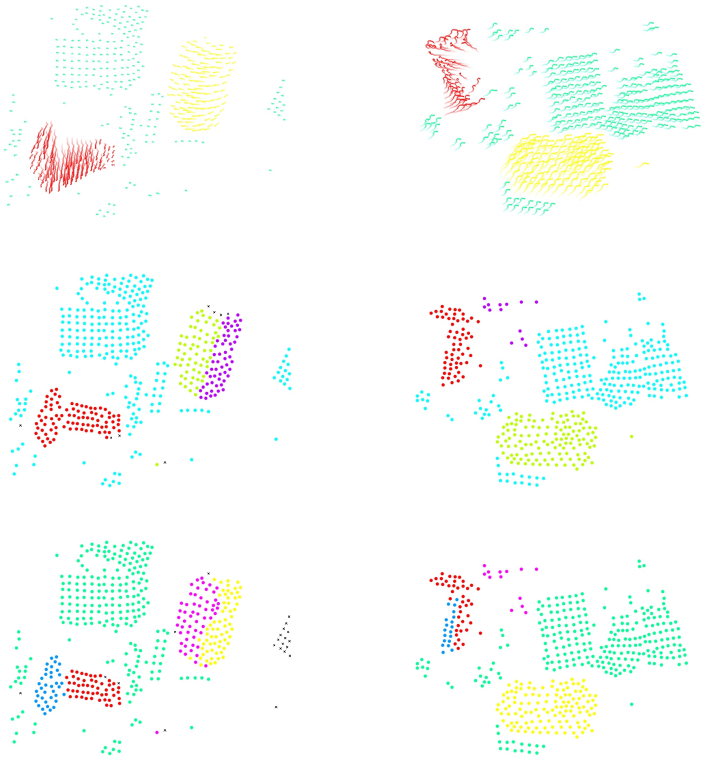


Figure 7.3: Two sequences (one left, one right) from the `Hopkins 155` dataset with little noise, highly distinct independent motions and $N = 3$. Top row: ground-truth color coded trajectories. Rows 2-3: Segmentation results with $N_I = 4$ and 5 respectively. Same labelling scheme as in Figure 7.3.

In summary, these results suggest that when the various motions in the sequence are very distinct, selecting N_I too small can result in having only N_I motions segmented correctly, with the other motions labeled as outliers. Conversely, when N_I is too large, the observed result is a strict over-segmentation of the correct segments. Our perception is that in an algorithm where N_I is specified *a priori*, this behavior is ideal. Unfortunately, the algorithm does not always behave ideally, especially when the underlying independent motions are less distinct, as will be described next.

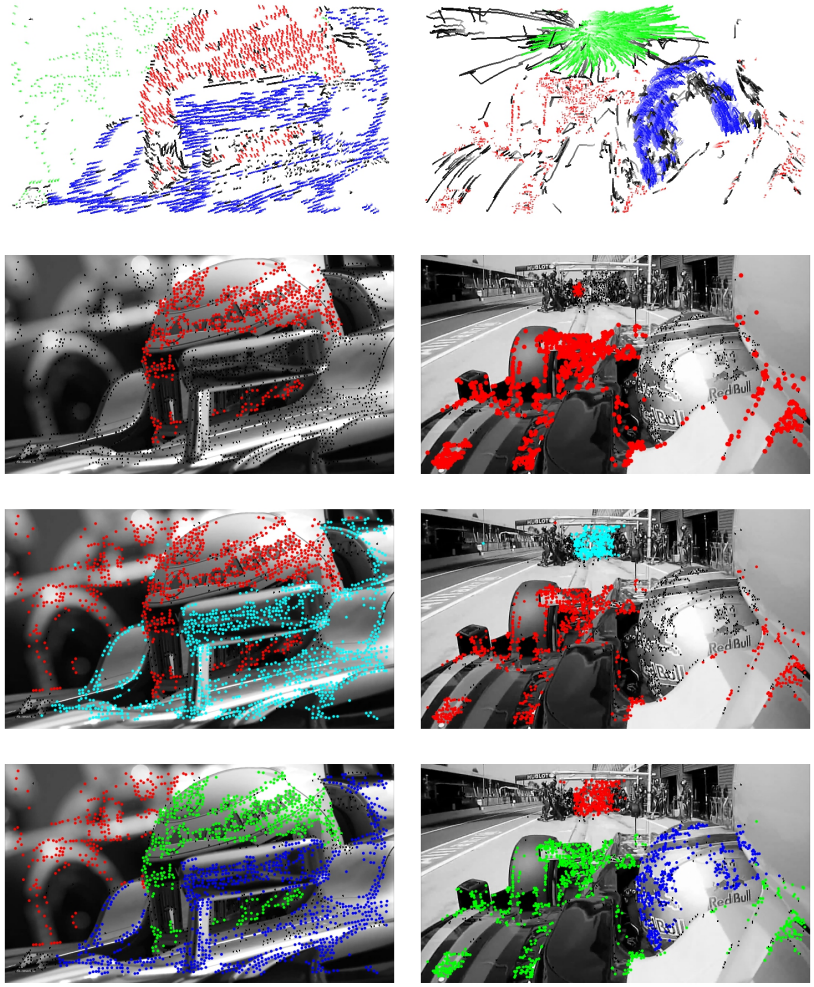


Figure 7.4: Two sequences from the `Formula 1` dataset with a large presence of outliers and trajectory noise. Top row: color coded trajectories using the results of our algorithm with $N_I = 3$. Rows 2-4: Segmentation results with $N_I = 1$ and 2 respectively. Segmentation results with $N_i = 3$ are shown in Row 4 for reference only. Same labelling scheme as in Figure 7.3.

Rigid sequences with noise and outliers

Under- and over-segmentation results for two real-world sequences from the `Formula 1` dataset are available in Figures 7.4 and 7.5, respectively. An off-the-shelf tracking algorithm is used to compute trajectories for these sequences, leading to a fair number of outliers and the presence of noise. There is no ground truth available for these sequences, but visual inspection reveals that

there are $N = 3$ independently moving objects in each of them. As with the previous example, the under-segmentation results are semantically relevant and mostly positive. We hypothesize that these good results are possible because the instantiation stage is very robust to outliers, and the model selection stage is designed to choose model combinations that render accurate predictions. The effectiveness of the outlier class at capturing trajectories that are not explained by any of the inlier models is also highlighted in these examples, including those trajectories from legitimate independently moving objects that remain unexplained because of using $N_I < N$.

There are two less satisfactory results in these examples that are worth mentioning. The first one is on the sequence shown to the right, for $N_I = 1$ where some of the background trajectories are labeled as inliers of the model that is used to explain the motion of the car. We observed that the inlier-labeled background trajectories have fortuitously negligible motions, and hypothesize that this (lack of) motion matches the motionlessness of the trajectories on the car itself (which has no relative motion with respect to the camera). The problem reliably disappears when 3 motions are used, suggesting that an extra independent motion does explain the trajectories from that background cluster of people better than the car's motion model. The second unsatisfactory result shown in this figure occurs on the sequence shown to the left, when $N_I = 2$, where the background trajectories are merged with the helmet ones. The hypothesis here is that the union of the subspaces of these two degenerate models fits within the effective rank of the subspace of a non-degenerate motion and does so with a reasonably small magnitude of the noise. That is, unlike the example of Figure 7.3, the different motions shown here are not as clearly separated.

The cases where $N_I > N$ are shown in figure 7.5. The resulting over-segmentations are primarily over-segmentations of the best segmentation result (with $N_I = 3$). This is not surprising, given that the trajectories are noisy and the algorithm is choosing motions that minimize the variance of the residuals as well as the modelling overlap (*i.e.*, the number of trajectories that are correctly explained by more than one model).

Finally, Figure 7.6 shows results from the sequence `2manko5` from the `Collective Motion` dataset [3]. This sequence shows two groups of people as they walk in opposite directions on a crosswalk, as well as some stationary objects. Based on this description we take $N = 3$ to be the appropriate

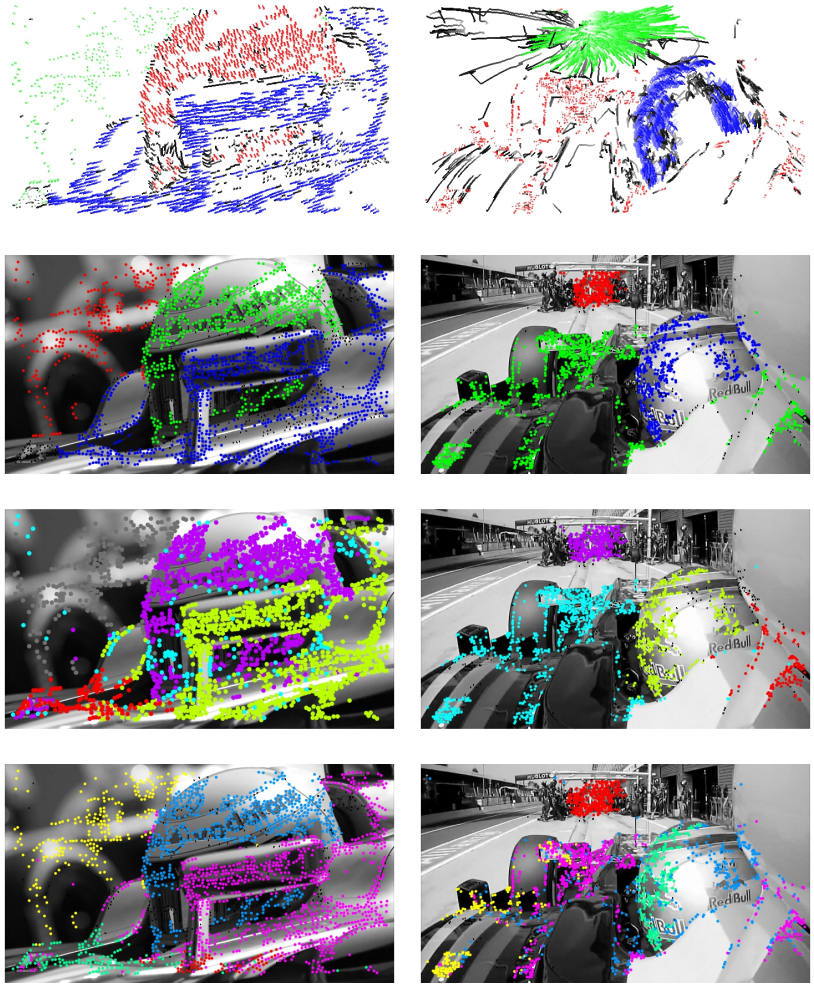


Figure 7.5: Two sequences from the `Formula 1` dataset with a large presence of outliers and trajectory noise. Top row: color coded trajectories using the results of our algorithm with $N_I = 3$. Row 2: Segmentation results with $N_I = 3$, shown for reference only. Rows 3-4: Segmentation results with $N_I = 4$ and 5 respectively. Same labelling scheme as in Figure 7.3.

number of motions in this scene. The results are very good for both of the under-segmentation examples but the over-segmentations are not clearly identifying additional salient motions. Still (as with the majority of the previous over-segmentation results) the (qualitatively) correct segmentation shown for $N_I = 3$ can be recovered by merging the trajectories from the green, red and purple classes from the $N_I = 5$ result, which is a desirable property if

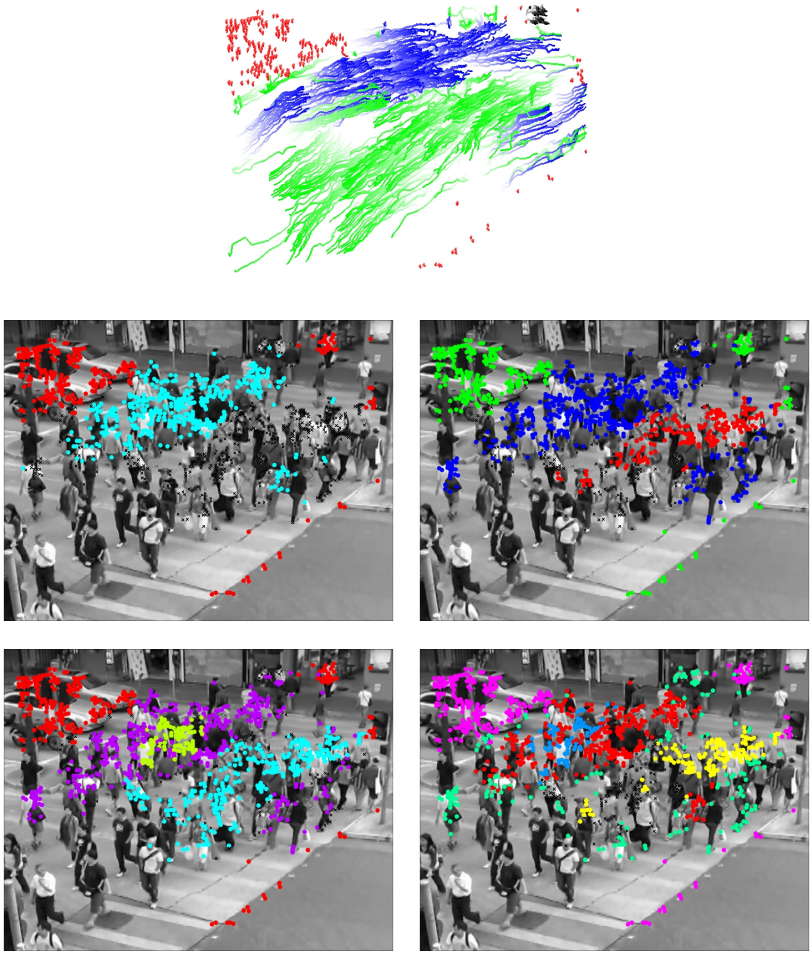


Figure 7.6: The `2manko5` Sequence from the `CM` database. Top: Trajectories, color-labeled by our algorithm using $N_I = 3$. Rows 2-3: Segmentation results with $N_I \in \{2, 3, 4, 5\}$.

one must over-segment. However, for $N_I = 5$, the segmentation appears to overlap multiple segments from the $N_I = 3$ result. We hypothesize that this is due to the large degree of model error that is present when fitting rigid motion models to a sequence with non-rigid objects.

Deviations From Rigidity

The main algebraic principle that supports most rigid motion segmentation algorithms is that the zero-mean, noise-free, observation data of a non-degenerate rigidly moving object spans a subspace of three dimensions (two for planar objects or for some cases of restricted camera motions or rotations) [24]. The importance of this finding is highlighted by the increased attention given to the motion segmentation problem after the result was published. In fact, more robust methods explicitly allow for small deviations from this assumption in order to incorporate some of the nuisances associated to real trajectory data.

One of the largest difficulties associated with the non-rigid motion segmentation problem originates precisely from the loss of this rigidity constraint. This makes individual motions difficult to characterize, due to the uncertainty associated with the necessary capacity of each motion or deformation model, as well as the ambiguity in determining when a non-rigid object is better represented as a single deforming one, or as several independently moving ones with simpler deformations, or as small-sized locally-rigid ones, as they do in [43].

The goal of this section is to briefly evaluate the proposed method on a series of non-rigid datasets to gain some understanding about its performance and to potentially identify some interesting research directions on the problem of non-rigid motion segmentation. The image sequences used here come from the `Hopkins 155` dataset introduced in [49], the `Collective Motion` database (hereby referred to as the `CM` database) introduced in [3], from the non-rigid structure from motion literature, including the `Paper` sequence [51], as well as the `Two cloths` and the `Tear` sequences [43] and from our own database (the `Formula 1` dataset). Tracks for all these sequences were computed using the standard KLT feature tracking algorithm [1]. For the `Hopkins 155` we used the trajectories provided in the dataset. Results from these experiments are shown in Figures 7.7 to 7.11.

Figure 7.7 shows trajectories that originate from a crowd of people running around a U-shaped path. The sequence is 88 frames long. Trajectories follow runners that appear in a close-to-planar (*i.e.*, degenerate) configuration, and the background trajectories do not move, which leads to a degenerate motion model. However, most of the non-static trajectories in this sequence display a

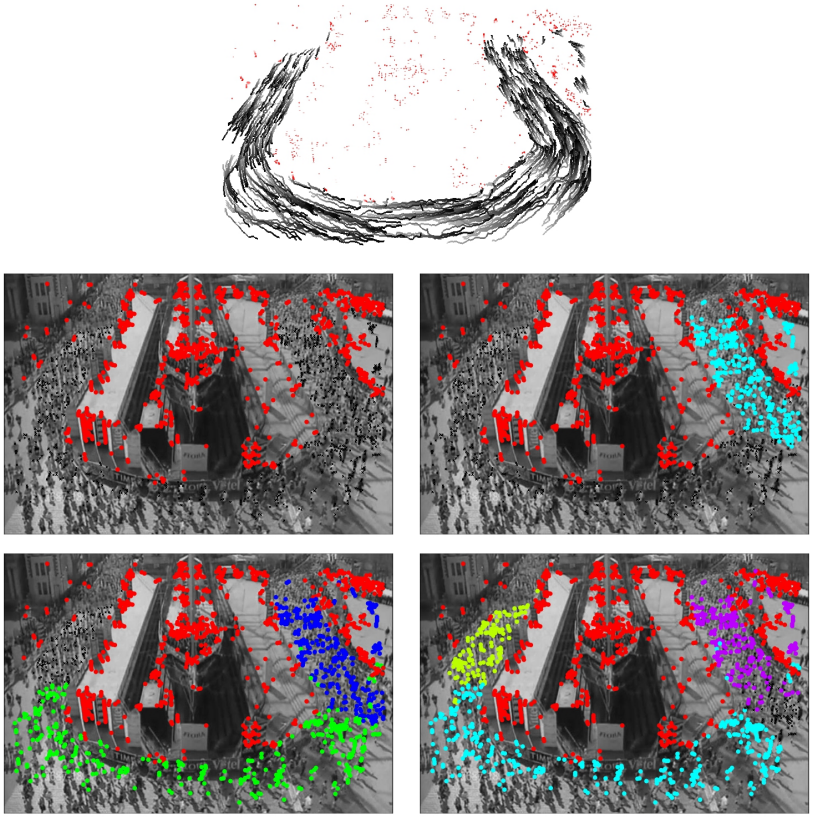


Figure 7.7: The `Marathon Round 2` sequence from the `Collective Motion` database. Top: Static versus non-static trajectories (as labeled by our algorithm when $N_I = 1$). Middle and Bottom Rows: Segmentation results with $N_I \in \{1, 2, 3, 4\}$.

locally rigid motion, with the exception of the group of trajectories that track people as they enter and exit the curved section of the path. Nonetheless, the algorithm successfully finds meaningful subsets of locally rigid trajectories, including for all of the under-segmentation cases (where $N_I \in \{1, 2, 3\}$).

Figure 7.8 shows segmentation results from the `Paper` sequence, where a piece of paper is bent from its original planar configuration into a shape that is concave towards the camera. The deformation is almost all in the vertical axis. This sequence also exhibits strong perspective effects. The results show how the algorithm approximates the surface using subsets of locally rigid surfaces that align with the direction of less deformation (vertical), and

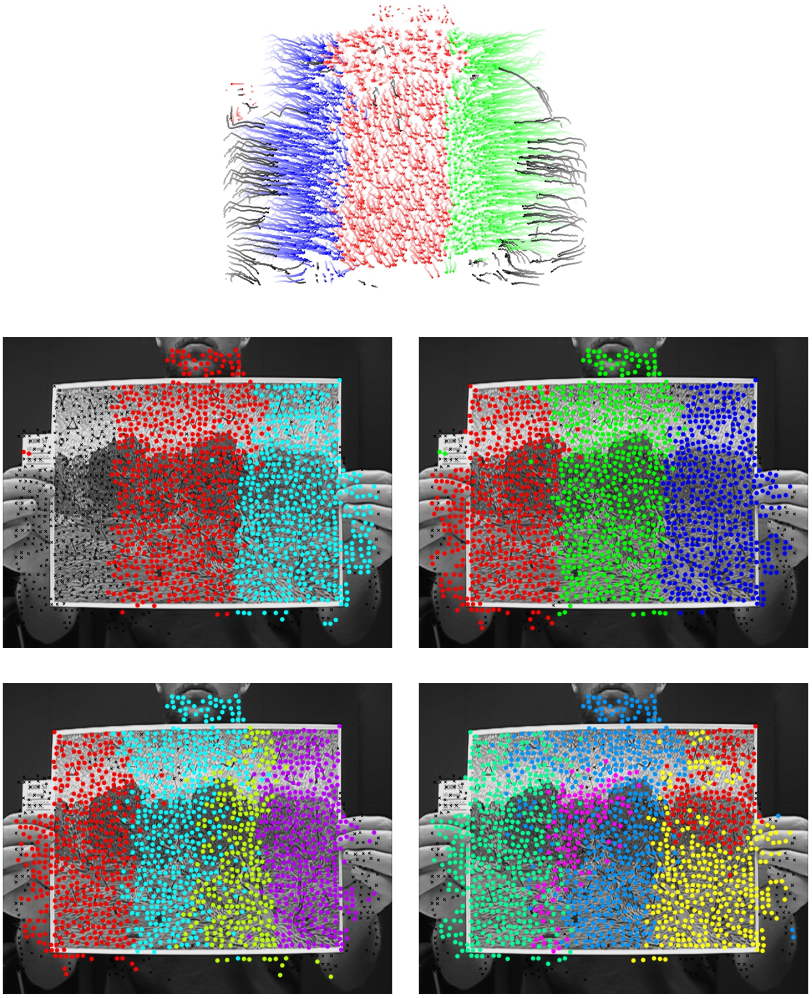


Figure 7.8: The `Paper` sequence. Top: Trajectories (color-labeled by our algorithm). Middle and Bottom Rows: Segmentation results for $N_I \in \{2, 3, 4, 5\}$.

break frequently in the direction of largest deformation (horizontal). The segmentation result of $N_I = 5$ initially suggests that the accuracy gains that result from increasing the number of models from 4 to 5 are only due to slightly better modelling of the noise, but careful inspection, motivated by the repeatability and reliability of this result, revealed that in fact the top right corner of the paper moves slightly differently from the bottom right part.

Figure 7.9 shows perfect segmentation results for each of the two indepen-

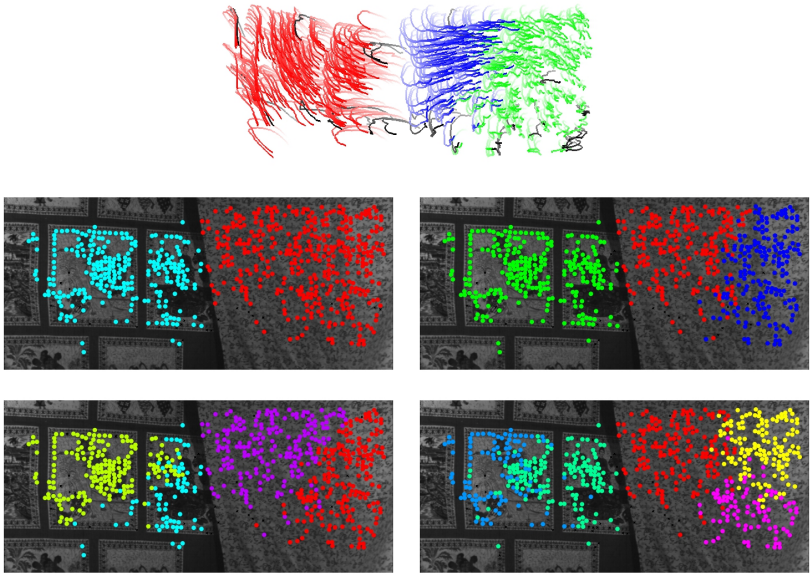


Figure 7.9: The `Two Cloths` sequence. Top: Trajectories (color-labeled by our algorithm). Middle and Bottom Rows: Segmentation results for $N_I \in \{2, 3, 4, 5\}$.

dently moving objects (when $N_I = 2$) of the `Two Cloths` sequence, despite their non-rigid nature. The images are of a thick tablecloth and a thin scarf as the wind moves them. To understand how a rigid model is good at describing the motion of non-rigid trajectories we plotted the model's predictions and the data observations in the same frame. These plots, shown in Figure 7.10, suggest that the motion of each cloth could be reconstructed with a synthetic non-planar surface by over-fitting the depth coordinate and the affine projections. Increasing the number of available motions further increases the accuracy of the reconstructions, particularly because the model-selection function chooses motion models where the largest accuracy gains are found.

Figure 7.11 shows results from a sequence where the motion of all trajectories is almost globally rigid for approximately one third of the frames (the first 85 out of 250) and only after do the motions separate. This sequence is a good example where the estimate of the number of independent motions by a human operator is biased due to perceptual grouping. Before running the algorithm our expectation of the result consisted in a left group and a right group. In reality, the motion difference between the left sleeve and the

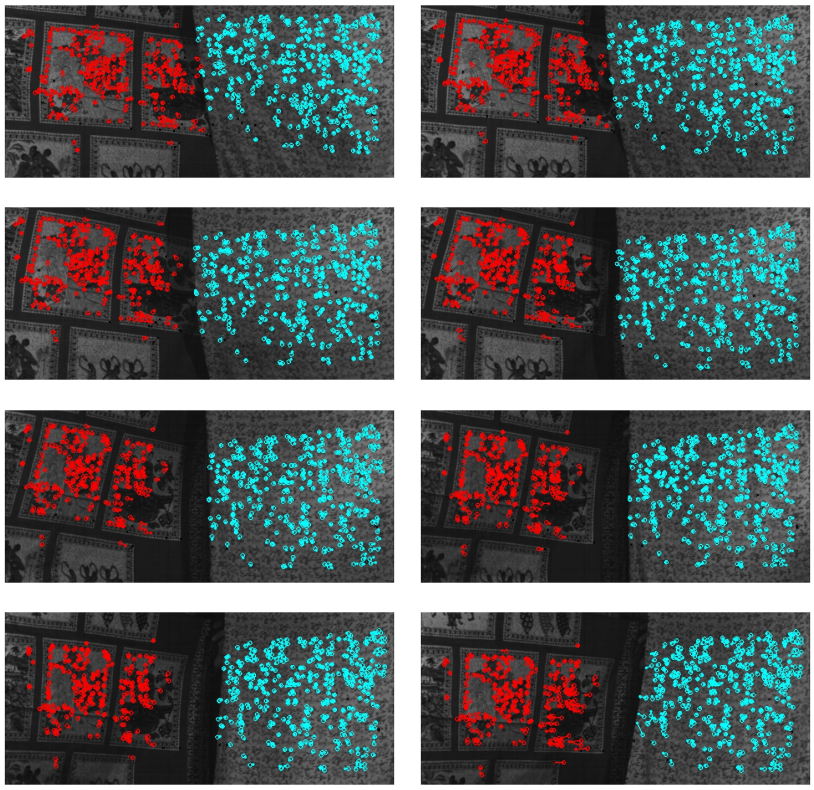


Figure 7.10: Observations and predictions from the `Two Cloths` sequence when motion-segmented with $N_I = 2$ for frames 6, 12, 18, 24, 30, 36, 42 and 48. Dots are the trajectory data, circles are the model predictions. Color coding shows the resulting labeling from our algorithm.

left hand of the volunteer is sufficiently salient to prevent them from being explained by the same motion model, even when the number N_I is restricted to two. Also, the trajectories on the piece of paper to the right of the image are too sparse and too noisy to encourage the objective function to include a model that explains them, even for large values of N_I.

7.3 Estimating the Number of Independent Motions

One of the first formal studies of problem of estimating the number (N) of independent motions in a scene was presented by [24]. The author rephrased the problem as the one of estimating the number of dimensions spanned by

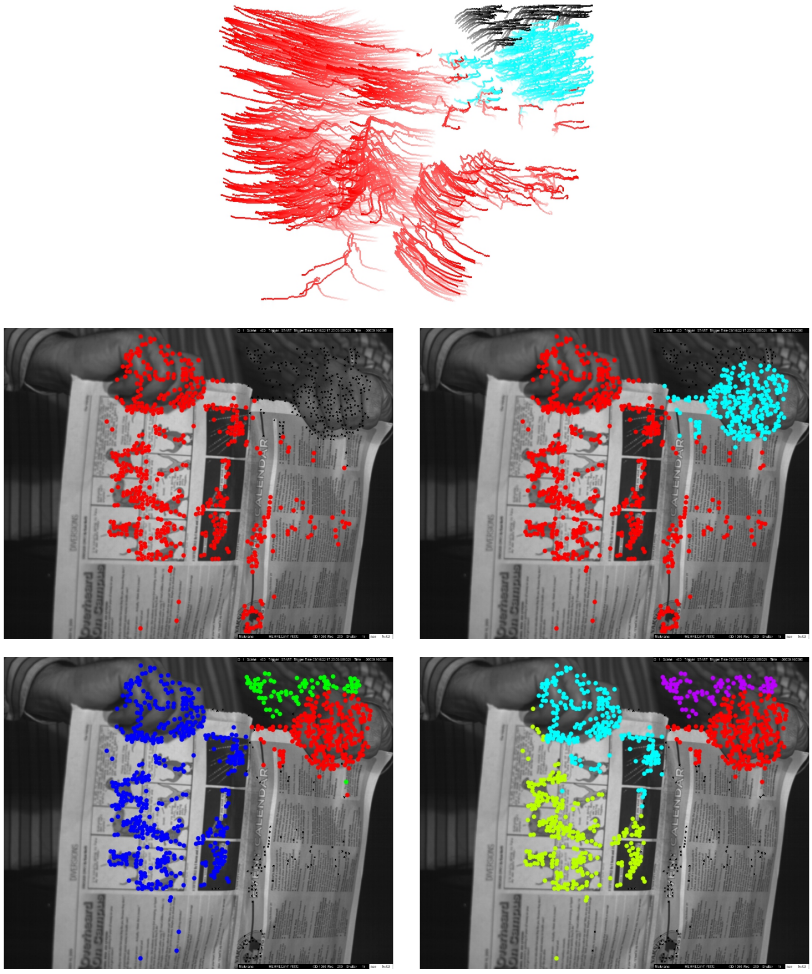


Figure 7.11: The `Tear` sequence. Top: Trajectories (color-labeled by our algorithm). Middle and Bottom Rows: Segmentation results for $N_I \in \{1, 2, 3, 4\}$.

the trajectory data (or equivalently, the rank of the observation matrix **W**).

The method responds to the simple algebraic notion that indicates that when using a subspace of limited dimensions to approximate a noisy observation matrix **W**, one can always get better approximations by increasing the dimensionality of the subspace. This suggests that an estimate for the size of the subspace must be regularized, or the resulting rank will always be the smallest matrix dimension. Kanatani uses the information theory framework to implement this regularization.

The proposed approach to estimate N is based on either of three geometric information criteria: the geometric Akaike Information Criterion [23], the geometric Minimum Descriptor Length [25] and what the authors call the Otsu-Ichimura Criterion, defined within the paper. All of these criteria are model selection functions that when minimized, aim to find the right balance between the magnitude of the data that is left unexplained by a subspace of limited dimensionality, and the cost of increasing the dimension of the subspace.

The problem with Kanatani's method is that the results can only be good when the estimates for the magnitudes of the noise are close to the real value (a problem that often has equal difficulty as the estimation of N itself) and when all of the trajectories have noise of similar magnitude. In addition, the method only finds the rank of the entire observation matrix, from which the estimate of the number of motions is still not trivially available. This difficulty arises from the potential presence of motion degeneracies or motion dependencies, which allow a larger number of motions to fit within a subspace of a given size, compared to the number of non-degenerate independent motions that would fit within a similarly sized subspace.

In contrast, in a much earlier paper by [4] the issue of estimating N is included as one of the necessary stages of the proposed MS algorithm, and the authors thoroughly detail the effects of motion dependency and motion degeneracy towards the effective rank of the observation matrix, but in the end, a manually set threshold on the magnitude of the unexplained residuals is used to determine the rank of $\mathbf{W}$.

In [54] the authors propose a matrix factorization method for sequences that include perspective effects, and they also aim to estimate N, but likewise, they minimize the problem of estimating N to simply doing

$$N = \frac{rank(\mathbf{W})}{6}, \tag{7.1}$$

citing [24] as a way to estimate the rank of the observation matrix.

A more modern approach to MS is introduced in [60] and the authors include a method to estimate N, which is an improvement over the model selection procedure of Kanatani [24]. They acknowledge that the use of a manually defined threshold to estimate the rank of $\mathbf{W}$ limits the applicability of the algorithm to sequences with a previously known noise magnitude. The

MS method is described in some detail in Section 2.2.2, but the estimate of N is briefly explained here. Assume the $2F \times N$ matrix $\mathbf{W}_r$ is the rank r limited reconstruction of $\mathbf{W}$. Then assume A_r is the $N \times N$ affinity matrix for each pairwise combination of trajectories from $\mathbf{W}_r$. The entry a_{ij} of A_r is evaluated using a negative exponential of the principal angle similarity between $\mathbf{w}_i$ and $\mathbf{w}_j$. The estimated $rank(\mathbf{W})$ is the one that maximizes the entropy of the affinity matrix

$$r^\star = \underset{r}{\text{argmax}} \sum_x A_r(x) log A_r(x). \tag{7.2}$$

The intuition is that when the rank is underestimated, trajectories are forced into a smaller subspace making their principal angles artificially similar, even for trajectories of independently moving objects. When the rank is overestimated, the reconstructed $\mathbf{W}_r$ is contaminated by some basis from the null space of $\mathbf{W}$, artificially decreasing the principal-angle affinity between trajectories from the same class. Only when the rank estimate is optimal do trajectories from the same class have maximum affinity, and trajectories from different classes minimum, and in this case the entropy of A_r is maximum [60]. The authors report correct estimations on 70.97% of sequences from the `Hopkins 155` dataset.

It seems clear that most previous attempts to globally estimate N (like those based on estimating the rank of the whole observation matrix $\mathbf{W}$, or the one described in the previous paragraph) are failing to acknowledge some critical aspects of this problem. In particular, that degenerate and non-degenerate motions have subspaces of different sizes, and that when dependent motions are present, at least one of the dimensions of the subspace is shared, artificially reducing the size of the combined subspace. Another missing aspect is that the magnitude of the noise is often different between the trajectories from different independently moving objects, either because of the raw image contents (motion blur, contrast, texture, etc.) or because of the uneven effect of unaccounted image formation phenomena (like perspective effects, or barrel distortion). The point is that the expected distance between much noisier trajectories and their appropriate low-dimensional subspace is therefore larger, and indeed can increase the difficulty of estimating the subspace itself, or even it's dimension.

We propose a solution that is similar in spirit to Kanatani's, where the

goal is to find the best balance between the optimal number of motions and the magnitude of the residuals, but we do it by quantifying the penalty for each motion individually, according to the estimated complexity of each independent motion, and calibrating the residual penalties using each motions' estimate of the magnitude of the noise.

To achieve this goal, our method finds the optimal number of motions by simply running the entire MS algorithm using $N_I \in \{1, 2, \ldots\}$. This means finding the optimal N-motion model combination ($\mathbb{T}_N$) for each value of N_I. Note that a model combination implicitly defines a labeling as well. The estimate for the number of motions corresponds to the value of N, that minimizes the function of Equation 5.10 across all $\{\mathbb{T}_N\}$.

Formally, this equates to solving the following optimization problem:

$$N^\star = \underset{N \in \{1,2,\ldots\}}{\operatorname{argmin}} \; \mathcal{O}(\mathbb{T}_N), \tag{7.3}$$

given an model combinations $\{\mathcal{M}_N\}$, where $\mathcal{O}(\cdot)$ corresponds to the same loss function of Equation 5.10, reproduced here for convenience:

$$\begin{aligned}\mathcal{O}(\mathbb{T}_N) = &\sum_{n=1}^{N}\sum_{p=1}^{P} l_{p,n} E\left(\mathbf{w}_p, \mathbf{M}_{j_n}\right) \\ &+ \lambda_\Phi \sum_{i=1}^{P} \Phi(\mathbf{w}_p, \mathbb{M}_j) + \lambda_\Psi \sum_{n=1}^{N} \Psi(\mathbf{M}_{j_n}).\end{aligned}$$

The intuition is that the minimum of the loss function $\mathcal{O}$ across different good model combinations of different sizes corresponds to the best balance between the magnitude of the error residuals (first term), and the penalty associated to the complexity of the models that are allowed and used represent each independent motion (the $\Psi(\mathbf{M})$ term). The function also penalizes explaining the same trajectory more than once. Favorable experimental results support this intuition.

The proposed method was tested on the `Hopkins 155` dataset, which contains sequences with $N = 2$ and $N = 3$ motions. The results are summarized in Table 7.1. Our method outperforms all other methods by a significant margin. To reduce the effects of result variability, ten model combinations were estimated for each of the values $N_I \in \{1, 2, 3, 4\}$, and an average objective function $\bar{\mathcal{O}}_{N_I}$ was computed for each N_I. The estimated N was the one

Table 7.1: Mean absolute error and accuracies for the estimated number of models. Our method outperforms all prior art by a significant margin.

	Method	$\mu(\lvert\text{error}\rvert)$	% Correct Estimation
2 Motions	ALC	1.13	30.00
	ELSA	0.33	75.00
	A-ASA	0.39	70.00
	Ours	**0.108**	**89.17**
3 Motions	ALC	1.25	11.43
	ELSA	0.49	57.14
	A-ASA	0.51	57.14
	Ours	**0.086**	**94.29**
Whole Database	ALC	1.16	25.81
	ELSA	0.37	70.97
	A-ASA	0.42	67.10
	Ours	**0.103**	**90.32**

associated to the minimum $\bar{\mathcal{O}}_{N_l}$.

It is worth noting that when the loss function is used to determine the optimal number of motions, the parameters $\lambda_\Phi = 1.5 \times 10^4$ and $\lambda_\Psi = 4.7 \times 10^1$ are different with respect to those used when doing motion segmentation ($\lambda_\Phi = 1.0 \times 10^5$ and $\lambda_\Psi = 1$). Both sets of parameters were determined empirically. Still, we argue that these two sets of parameters make intuitive sense, given that the segmentation problem benefits from more discriminative labelling (ambiguous labeling is penalized by λ_Φ, which is 1 order of magnitude bigger when doing segmentation) and the estimation of N problem benefits from more efficient use of the available representation resources (the model complexity term is 50 times bigger when estimating N).

Figure 7.12 shows the distribution of the discrepancies between the estimated number of motions using our method, and the ground truth. From this histogram it is clear that the majority of the errors are under-estimations of the number of motions. Visual inspection reveals that the majority of the sequences with under-estimated numbers of motions contain two or more degenerate motions, and in many cases, these motions are dependent. These sequences are really difficult since the rank of the observation matrix $\mathbf{W}$ of trajectories from 2 dependent degenerate motions is $rank(\mathbf{W}) \leq 3$ which can be modeled by a single non-degenerate affine motion, and especially when depths can be over-fit to approximate the data. Figure 7.13 shows ground-

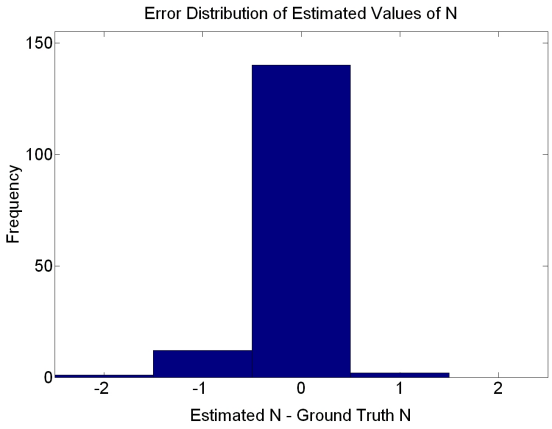


Figure 7.12: Histogram of the discrepancy between the estimated N and the ground truth. Only two sequences are over-estimated. The rest of the errors are under-estimations.

truth, color-coded trajectories of six of the under-estimated examples. In these cases, a spatial coherence prior (that includes coherence on relative depth) may be a way to increase the accuracy of the estimation.

In contrast, Figure 7.14 shows the only two over-estimated examples where we hypothesize that the large range of true depths, and possibly the existence of perspective effects, may have led to the over-estimation of N.

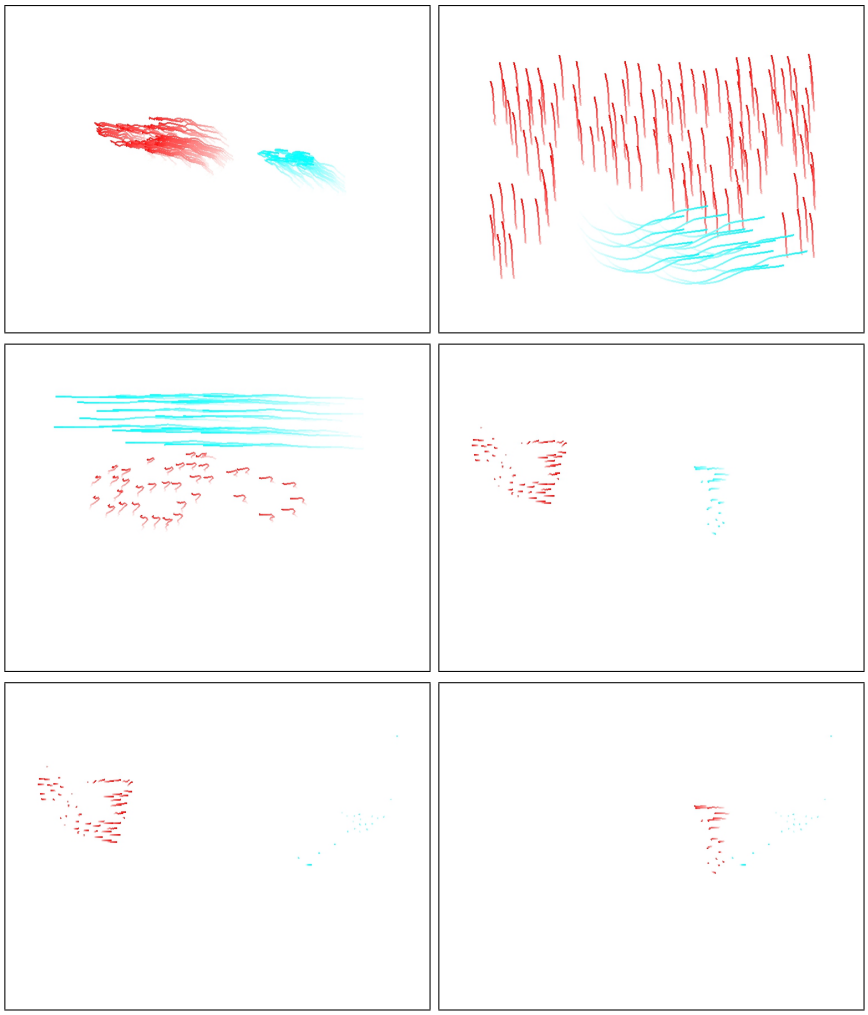


Figure 7.13: Trajectories from some examples of sequences where the estimated number of motions is $N = 1$ while the correct value is $N = 2$. Color indicates ground truth motion segments.

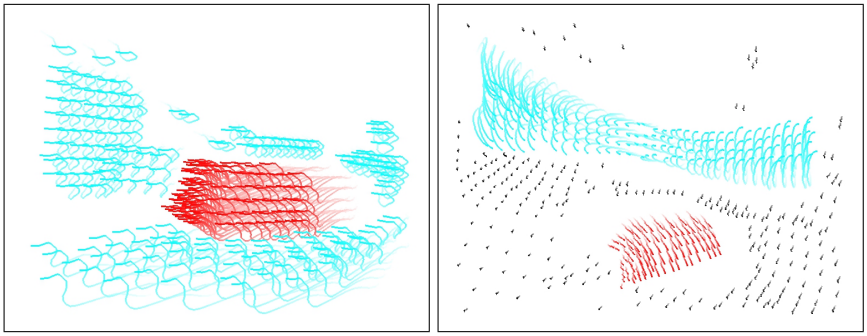


Figure 7.14: Trajectories from the two sequences where the estimated number of motions is $N = 3$ (left) and $N = 4$ (right) while the correct values are $N = 2$ (left) and $N = 3$ (right). Color indicates ground truth motion segments.

References

[1] http://www.ces.clemson.edu/~stb/klt. 111

[2] Blake, R., and Shiffrar, M. Perception of Human Motion. *Annual Review of Psychology 58*, 1 (2007), 47–73. 1

[3] Bolei, Z., Xiaoou, T., and Xiaogang, W. Measuring crowd collectiveness. In *CVPR* (2013), IEEE, pp. 3049–3056. 23, 108, 111

[4] Boult, T. E., and Brown, L. G. Factorization-based segmentation of motions. In *Workshop on Visual Motion* (1991), pp. 179–186. 12, 117

[5] Bregler, C., Hertzmann, A., and Biermann, H. Recovering non-rigid 3D shape from image streams. In *CVPR* (2000), pp. 690–696. 12

[6] Briassouli, A., and Ahuja, N. Integrated spatial and frequency domain 2D motion segmentation and estimation. In *ICCV* (2005), vol. 1. 20

[7] Buchanan, A., and Fitzgibbon, A. Combining local and global motion models for feature point tracking. In *CVPR* (June 2007), pp. 1–8. 22

[8] Costeira, J., and Kanade, T. A multi-body factorization method for motion analysis. Tech. Rep. CMU-CS-TR-94-220, CMU - Computer Science Department, Pittsburgh, PA, September 1994. 1, 11, 25

[9] Costeira, J., Kanade, T., and Invariants, M. A. A multi-body factorization method for independently moving objects. *IJCV 29* (1998), 159–179. 11, 12, 25

[10] Elhamifar, E., and Vidal, R. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009). 14, 33, 98, 102

[11] Fan, Z., Zhou, J., and Wu, Y. Multibody motion segmentation based on simulated annealing. *CVPR 1* (2004), 776–781. 13

[12] Fischler, M. A., and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* (1981). 28, 36, 61

[13] Furukawa, Y., and Ponce, J. Accurate, dense, and robust multi-view stereopsis. In *CVPR* (2007), pp. 1–8. 23

[14] Gilaie-Dotan, S., Saygin, A. P., Lorenzi, L. J., Egan, R., Rees, G., and Behrmann, M. The role of human ventral visual cortex in motion perception. *Brain 136*, 9 (2013), 2784–2798. 1

[15] Goh, A., and Vidal, R. Segmenting motions of different types by unsupervised manifold clustering. In *CVPR* (2007). 17

[16] Gruber, A., and Weiss, Y. Factorization with uncertainty and missing data: Exploiting temporal coherence. In *NIPS* (2003). 15, 27

[17] Gruber, A., and Weiss, Y. Multibody factorization with uncertainty and missing data using the EM algorithm. In *CVPR* (2004), vol. 1, pp. I–707–I–714 Vol.1. 15, 27

[18] Gruber, A., and Weiss, Y. Incorporating non-motion cues into 3D motion segmentation. *CVIU 108*, 3 (2007), 261–271. 15, 27

[19] Huttenlocher, D., and Ullman, S. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision 5*, 2 (1990), 195–212. 29

[20] Irani, M., and Anandan, P. Parallax geometry of pairs of points for 3d scene analysis. *Computer Vision—ECCV'96* (1996). 29

[21] Jepson, A. D., Fleet, D. J., and El-maraghi, T. F. Robust online appearance models for visual tracking. pp. 1296–1311. 1

[22] Jia, H., and Martinez, A. M. Low-rank matrix fitting based on subspace perturbation analysis with applications to structure from motion. *PAMI 31*, 5 (2009), 841–854. 21, 30

[23] Kanatani, K. Geometric information criterion for model selection. *IJCV 26*, 3 (1998), 171–189. 21, 29, 31, 117

[24] Kanatani, K. Motion segmentation by subspace separation: Model selection and reliability evaluation. *Int J Image Graphics* (2002). 103, 111, 115, 117

[25] Kanatani, K., and Matsunaga, C. Geometric mdl and its media applications. In *Proc. 2000 Workshop Informationbased Induction Science* (2000), pp. 45–51. 117

[26] Kumar, M., Torr, P., and Zisserman, A. Learning layered motion segmentations of video. In *ICCV* (Oct. 2005), vol. 1, pp. 33–40 Vol. 1. 20, 28

[27] Lakdawalla, A., and Hertzmann, A. Shape from Video: Dense Shape, Texture, Motion and Lighting from Monocular Image Streams. In *Proceedings of the First International Workshop on Photometric Analysis For Computer Vision* (Rio de Janeiro, Brazil, 2007), p. 8 p. 16, 22

[28] Laptev, I., Belongie, S. J., PÃl′rez, P., and Wills, J. Periodic motion detection and segmentation via approximate sequence alignment. *ICCV 1* (2005), 816–823. 19, 28

[29] Li, H. Two-view motion segmentation from linear programming relaxation. In *CVPR* (2007). 19, 28, 31

[30] Llado, X., Del Bue, A., and Agapito, L. Non-rigid 3D factorization for projective reconstruction. In *BMVC* (Sept 2005). 12

[31] Lucas, B. D., and Kanade, T. An iterative image registration technique with an application to stereo vision. pp. 674–679. 1, 2

[32] Ma, Y., Derksen, H., Hong, W., and Wright, J. Segmentation of multivariate mixed data via lossy data coding and compression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*, 9 (Sept 2007), 1546–1562. 14

[33] Newsome, W. T., and Park, E. B. A selective impairment of motion perception following lesions of the middle temporal visual area (mt. *Journal of Neuroscience 8* (1988), 2201–2211. 1

[34] Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS* (2001), MIT Press, pp. 849–856. 91

[35] Rao, S., Tron, R., Vidal, R., and Ma, Y. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (June 2008), pp. 1–8. 14

[36] Rao, S. R., Yang, A. Y., Wagner, A. W., and Ma, Y. Segmentation of hybrid motions via hybrid quadratic surface analysis. In *ICCV* (2005), pp. 2–9. 17

[37] Schindler, K., and Suter, D. Two-view multibody structure-and-motion with outliers. In *CVPR* (2005), pp. 676–683. 18, 22, 28, 29, 31

[38] Schindler, K., U, J., and Wang, H. Perspective *n*-view multibody structure-and-motion through model selection. In *ECCV (1)* (2006), pp. 606–619. 18, 19, 28, 29, 31

[39] Shi, J., Belongie, S., Leung, T., and Malik, J. Image and video segmentation: the normalized cut framework. *ICIP 1* (1998), 943. 22

[40] Shi, J., and Tomasi, C. Good features to track. pp. 593–600. 1, 2

[41] Sivic, J., Schaffalitzky, F., and Zisserman, A. Object level grouping for video shots. *IJCV* (April 2006). 22

[42] Sugaya, Y., and Kanatani, K. Multi-stage unsupervised learning for multi-body motion segmentation. In *IEICE Transactions on Information and Systems* (2004), pp. 1935–1942. 16, 29

[43] Taylor, J., Jepson, A. D., and Kutulakos, K. N. Non-rigid structure from locally-rigid motion. In *CVPR* (2010), pp. 2761–2768. 21, 111

[44] Tomasi, C., and Kanade, T. Detection and tracking of point features. Tech. rep., International Journal of Computer Vision, 1991. 1, 2

[45] Tomasi, C., and Kanade, T. Shape and motion from image streams under orthography: a factorization method. *IJCV 9*, 2 (1992), 137–154. 11, 28, 73

[46] Tong, W.-S., Tang, C.-K., and Medioni, G. Epipolar geometry estimation for non-static scenes by 4D tensor voting. *CVPR 1* (2001), 926. 19, 28

[47] Tong, W.-S., Tang, C.-K., and Medioni, G. Simultaneous two-view epipolar geometry estimation and motion segmentation by 4D tensor voting. *PAMI 26*, 9 (2004), 1167–1184. 28

[48] Torr, P. H. S. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London A 356* (1998), 1321–1340. 18, 21, 28, 29, 31

[49] Tron, R., and Vidal, R. A benchmark for the comparison of 3D motion segmentation algorithms. In *CVPR* (2007). 23, 111

[50] Ullman, S. Maximizing rigidity: The incremental recovery of 3-D structure from rigid and rubbery motion. *Perception 13* (1983), 255–274. 20

[51] Varol, A., Salzmann, M., Tola, E., and Fua, P. Template-free monocular reconstruction of deformable surfaces. In *ICCV* (2009), IEEE, pp. 1811–1818. 111

[52] Vasconcelos, N., and Lippman, A. Empirical bayesian motion segmentation. *PAMI 23*, 2 (2001), 217–221. 15, 27

[53] Vidal, R., and Hartley, R. Motion segmentation with missing data using powerfactorization and gpca. In *In CVPR* (2004), pp. 310–316. 17

[54] Vidal, R., Soatto, S., and Sastry, S. *A Factorization Method for 3D Multi-body Motion Estimation and Segmentation*. Memorandum (University of California, Berkeley, Electronics Research Laboratory). Electronics Research Laboratory, College of Engineering, University of California, 2002. 117

[55] Weiss, Y. Segmentation using eigenvectors: A unifying view. In *ICCV* (1999), pp. 975–982. 33

[56] Wolf, L., and Shashua, A. Two-body segmentation from two perspective views. In *CVPR* (December 2001), pp. 263–270. 16

[57] Wu, Y., Zhang, Z., Huang, T. S., and Lin, J. Y. Multibody grouping via orthogonal subspace decomposition. *CVPR 2* (2001), 252. 13

[58] Yan, J., and Pollefeys, M. Articulated motion segmentation using ransac with priors. In *Workshop on Dynamical Vision* (2005), vol. 4358, pp. 75–85. 19, 28

[59] Yan, J., and Pollefeys, M. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV, Part IV* (2006), vol. 3954, Springer, pp. 94–106. 13, 14, 28, 29, 33

[60] Zappella, L., Lladó, X., Provenzi, E., and Salvi, J. Enhanced Local Subspace Affinity for feature-based motion segmentation. *Pattern Recognition* (2011). 14, 33, 117, 118

[61] Zappella, L., Provenzi, E., Lladó, X., and Salvi, J. Adaptive motion segmentation algorithm based on the principal angles configuration. *ACCV* (2011). 98

[62] Zelnik-Manor, L., and Irani, M. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. In *CVPR* (2003), pp. 287–293. 12